

Copyright

by

Benjamin Van Blarigan

2012

**The Thesis Committee for Benjamin Van Blarigan  
Certifies that this is the approved version of the following thesis:**

**Automated Estimation of Time and Cost for Determining Optimal  
Machining Plans**

**APPROVED BY  
SUPERVISING COMMITTEE:**

**Supervisor:**

---

Matthew I. Campbell

---

Wei Li

**Automated Estimation of Time and Cost for Determining Optimal  
Machining Plans**

**by**

**Benjamin Van Blarigan, B.S.**

**Thesis**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Science in Engineering**

**The University of Texas at Austin**

**May 2012**

## **Acknowledgements**

It is a pleasure to thank the many people who have made this thesis possible.

First and foremost I would like to thank my advisor Dr. Matthew Campbell. From the time he accepted me into his research group, to the submittal of this thesis, working with him has been an absolute pleasure. He has been a constant source of encouragement, sound advice, technical expertise, and great ideas.

I would like to thank Dr. Wei Li for electing to be a member of my thesis committee, and for all of his valuable input.

I wish to thank my family, particularly my father and mother, Peter and Tracy. My mother was my teacher for many years, and she granted me the academic base upon which I credit all of my academic success. My parents, both engineers, introduced me to a life of engineering at an early age, and I have loved it ever since.

Finally I thank my fiancé Ashley, for moving away from our lifelong home to fulfill my dream, for being a constant source of encouragement and support, and for tolerating the moodiness and late nights associated with those particularly busy weeks. I couldn't have done it without you.

## **Abstract**

# **Automated Estimation of Time and Cost for Determining Optimal Machining Plans**

Benjamin Van Blarigan, M.S.E

The University of Texas at Austin, 2012

Supervisor: Matthew I. Campbell

The process of taking a solid model and producing a machined part requires the time and skillset of a range of professionals, and several hours of part review, process planning, and production. Much of this time is spent creating a methodical step-by-step process plan for creating the part from stock. The work presented here is part of a software package that performs automated process planning for a solid model. This software is capable of not only greatly decreasing the planning time for part production, but also give valuable feedback about the part to the designer, as a time and cost associated with manufacturing the part. In order to generate these parameters, we must simulate all aspects of creating the part. Presented here are models that replicate these aspects. For milling, an automatic tool selection method is presented. Given this tooling, another model uses specific information about the part to generate a tool path length. A machining simulation model calculates relevant parameters, and estimates a time for machining given the tool and tool path determined previously. This time value, along

with the machining parameters, is used to estimate the wear to the tooling used in the process. Using the machining time and the tool wear a cost for the process can be determined. Other models capture the time of non-machining production times, and all times are combined with billing rates of machines and operators to present an overall cost for machining a feature on a part. If several such features are required to create the part, these models are applied to each feature, until a complete process plan has been created.

Further post processing of the process plan is required. Using a list of available machines, this work considers creating the part on all machines, or any combination of these machines. Candidates for creating the part on specific machines are generated and filtered based on time and cost to keep only the best candidates. These candidates can be returned to the user, who can evaluate, and choose, one candidate. Results are presented for several example parts.

## Table of Contents

Table of Contents .....	vii
List of Tables .....	ix
List of Figures .....	x
Chapter 1: Introduction .....	1
Chapter 2: Related Work .....	4
<b>SECTION 1: MANUFACTURING SIMULATION .....</b>	<b>6</b>
Chapter 3: Tool Selection .....	7
3.1. Offset Curves .....	7
3.2. Tool Type Selection .....	9
3.3. Offset Storage .....	11
3.4. Starting Face Assumptions .....	13
3.4.1. Geometric Requirements .....	13
3.4.2. Virtual boundary assumptions .....	14
3.5. Tool size selection .....	16
3.6. Limitations and Future Work .....	20
Chapter 4: Tool Path Planning .....	22
4.1. Primary Tool .....	22
4.2. Small Tool .....	27
4.2.1 Area removed by big tool .....	28
4.2.2 Finishing Distance .....	35
4.3. Total Tool Travel .....	38
Chapter 5: Machining Simulation .....	43
Tool Wear Modeling .....	47

Chapter 6: Non-Machining Production Time .....	53
Chapter 7: Determining Operation Cost .....	57
Chapter 8: Simulation Results .....	58
8.1. Internal Pocket Single Tool .....	58
8.2. Internal Pocket Double Tool.....	60
8.3. Exterior Pocket Double Tool .....	63
8.4 Adjacent Partitions.....	66
8.5. Conclusions and Future Work .....	69
<b>SECTION 2: MACHINING PLAN PROCESSING.....</b>	<b>70</b>
Chapter 9: Machine Selection.....	72
Chapter 10: Machining Options.....	78
Chapter 11: Machining Plan Simplification .....	80
Chapter 12: Candidate Generation.....	87
Chapter 13: Candidate Filtering.....	90
Chapter 14: Process Planning Results.....	92
14.1. Stepped Part .....	92
14.1.1 3 Operations, 3 Fixtures.....	93
14.1.2. 2 Operations, 2 Fixtures.....	96
14.1.3. 1 Operation, 1 Fixture.....	97
14.2. “S” Part .....	99
14.2.1 Fillets First .....	100
Chapter 15: Discussion and Future Work.....	106
Bibliography .....	108



## List of Tables

Table 1: Format of the Shapedata Array .....	12
Table 2: Relevant Tooling Parameters.....	20
Table 3: Part Parameters Relevance for Machine Selection .....	73
Table 4: Required Parameters from Part and Machine for Machine Compatibility Check .....	73
Table 5: Details for Candidate Points for 3 Operations for Stepped Feature .....	94
Table 6: Machine Data Available to AMFA.....	95
Table 7: Details for Candidate Points for 2 Operations on Stepped Feature Part..	97
Table 8: Details for Candidate Points for 1 Operation on Stepped Feature Part ...	98
Table 9: “S” part, 8 Operations.....	102
Table 10 : Fast and Costly Cluster.....	103
Table 11: Slow and Cheap Cluster.....	104

## List of Figures

Figure 1: Examples of Offset Curves.....	7
Figure 2: Offset Curves of a Specified Radius for a Given Boundary Curve.....	8
Figures 3: Typical Tools Considered for use by AMFA .....	9
Figure 4: Typical Ball End Mills Considered for use by AMFA .....	10
Figure 5.a,b : Partitions that Require End Mills and Ball Mills .....	11
Figure 6: Conversion of Offset Curves to Shapedata Array .....	13
Figure 7: Virtual Sharp Corners on a Partition Mated with Another Partition .....	15
Figure 8: Maximum Tool Diameter for Accessibility .....	16
Figure 9: Chosen Tool Diameter for Machining.....	17
Figure 10: Boundary Tangent Tool Path .....	23
Figure 11: AMFA Toolpath Compared to Popular CAM Package Toolpath.....	25
Figure 12: Additional Path Length to Travel Between Offsets .....	26
Figure 13: Area Covered by Sweeping a Tool Along a Line .....	29
Figure 14. Areas of Overlap for Tracing a Square Boundary.....	30
Figure 15: X Versus the Number of Sides for Regular Polygons.....	31
Figure 16: Boundary Curve vs. Finishing Distance for Open Pocket.....	36
Figure 17: Model to Demonstrate Finishing Length Calculation .....	37
Figure 18: Negative of Solid Model Shown in Figure 17.....	37
Figure 19: Identifying Edges for Finishing Pass.....	38
Figure 20: Various Parts of Equal Volume and (Top) Surface Area.....	39
Figure 21: Extra Path Length for Travel Between Z-Steps .....	41
Figure 22: Machining Times vs. Cutting Speed .....	46
Figure 23: Cost Driven by Machine Occupation Time vs. Cutting Speed .....	47

Figure 24: Total Cost for Machining Processes Based on Cutting Speed .....	48
Figure 25: Tooling Cost vs. Cutting Speed.....	51
Figure 26: Total Cost of Performing Operation with Varying Cutting Speed.....	52
Figure 27: Flowchart of Fixture Time Assignment .....	54
Figure 28: Flowchart of Tool Change Time assignment .....	55
Figure 29: Flowchart of Machine Switch Time Assignment.....	56
Figure 30: Single Tool Internal Pocket Part Used for FeatureCAM and AMFA Comparison .....	58
Figure 31: AMFA Return for Single Tool Pocket .....	59
Figure 32: Two Tool Internal Pocket Part Used for FeatureCAM and AMFA Comparison.....	60
Figure 33: AMFA Return for Two Tool Pocket .....	62
Figure 34: Double Tool External Pocket Used for FeatureCAM and AMFA Comparison .....	63
Figure 35: AMFA Return for Two Tool Open Pocket .....	65
Figure 36: Adjacent Partitions Part.....	66
Figure 37: Tool Path Comparison for Adjacent Partitions Example .....	67
Figure 38: Single Partition Tool Path Comparison for Adjacent Partitions Example .....	68
Figure 39: Structure of a Machining Plan.....	70
Figure 40: Comparing Part Size with Machine Maximum Part Size.....	75
Figure 41: Performing an Operation on all Available Machines.....	78
Figure 42: Creation and Structure of a Machining Options List .....	79
Figure 43: “S” part .....	80
Figure 44: negative of “S” .....	81

Figure 45: Human generated process plan for “S” .....	82
Figure 46: AMFA process plan for “S” part.....	84
Figure 47: merging two operations .....	86
Figure 48: Generation of Candidate Points.....	88
Figure 49: Structure of a Candidate .....	88
Figure 50: Candidate Filtering by Removing Dominated Candidates.....	90
Figure 51: Stepped Feature Part.....	92
Figure 52: 3 Operations for Stepped Feature .....	94
Figure 53: 2 Operations for Stepped Feature .....	96
Figure 54: 1 Operation for Stepped Feature .....	98
Figure 55: Variety of Machining Plans for Stepped Feature Part.....	99
Figure 56: “S” part, 8 Operations .....	101
Figure 57: Machine-Specific Candidates for 3 Different Machining Plans .....	105

## **Chapter 1: Introduction**

The process of taking a solid model and producing a machined part requires the time and skillset of a range of professionals, and several hours of part review, process planning, and production. In the last decade, software has been able to ease some of the burden of the process, assisting with product cost estimation and machine operation. Automated software has not, however, been able to replace the process planning aspect of machined part production. Process planning is typically hailed as a “trick of the trade”, where machinists with decades of experience can perform this task quickly and efficiently, and less experienced employees make common oversights and errors. While the human reasoning power cannot be matched by software, in general it is believed that most decisions made by trained machinists can be mapped to concrete rules governing appropriate process planning. It is reasonable to assume, if these rules could be captured and translated to software code based on sound principles and engineering, the “mind of the machinist” could be replicated as a software package with the ability to perform process planning given only a solid model as an input, and the available tooling and machinery. The software could produce, for the users’ consideration, a process plan, along with the associated time and cost to perform that plan.

Developing such a piece of software is the goal of the DARPA iFAB program which has funded this research. Development has been carried out for the past 12 months, and will continue into the foreseeable future. Currently the package, named AMFA or Automated Manufacturing Feedback Analysis, is able to deliver the desired outputs for simple solids. In order to do this, all aspects of manufacturing a part must be represented and estimated. Presented in this work is the progress to date on modules of the software that perform machining simulations, estimate non-machining production

times, and predict costs for each associated stage of manufacturing. All of these parameters are useful in building a complete process plan for machining. Additionally, efforts to reason about the resulting process plan, and simplify it for the sake of the software operator will be presented.

AMFA operates on a solid model of a part in the STEP format (ap203 or ap214) [1]. The software is split into three modules, each with a distinct and unique function. The first module operates with the solid model, and uses open source CAD libraries to perform Boolean operations, decompose the negative of the solid into convex partitions, calculate parameters of each partition, and convert and store all information as a graph (.GXML file) [2].

The second module, referred to as “representation”, works with this graph. This module uses graph grammar rules to identify appropriate machining conditions for each partition. It will determine if a partition is accessible, and if so, select a machine type (i.e. vertical mill, drill press, etc.) with which to machine it, a fixture orientation, and a machine starting face; this module is described by Fu et al [3].

The third module, referred to as “evaluation”, is the subject of this work, and will be presented in detail. The first major task of this module is to analyze the removal of a single partition, where the type of machine, part orientation, and machining face, have been determined by the representation module. This requires a series of steps, similar to the process required for actual machining. First, the appropriate tooling must be selected. Next, the tool path must be planned. Finally, the operating parameters of the machine must be selected. The result is reporting a time to machine the given partition.

Building a machining process plan for a series of operations which involve machining a single partition requires a search algorithm, since there are several options for creating a process plan, by altering the order of partitions removed, tool access

directions, etc. The ultimate goal of evaluating a process is to return to the search process a metric that allows a guided search to evaluate the quality of the process. Not only will this allow the use of a guided search rather than a blind search process, but the metrics generated will cumulatively determine the “quality” of the overall process plan. For machining, it is clear that the needed metrics are the time required and the associated cost to produce a part using the process plan under consideration. While part quality would also be a desirable metric, the correlation between machining parameters and part quality has not been well characterized. Therefore quality metrics are left for future research, and will not be discussed here.

In many instances of manufacturing, the main driver of cost is the time required to complete the part. While the use of exotic materials or tooling can become dominant, typically the labor cost of the operator, the cost associated with machine use, and the hourly overhead of a manufacturing facility will greatly exceed the cost of raw materials and tooling consumed in the process [4]. Therefore, we can conclude with some assurance that to estimate the cost of creating a part, we need an accurate estimate of the amount of time required to make the part. The time can be broken down into several distinct tasks, which are machine setup, fixture setup and breakdown, machining time, part removal and relocation, and inspection and deburring. Assigning machining time has been discussed, and the remaining time elements will be discussed here.

At this time the software is applicable only to 3-dimensional milling processes, but will be expanded to include simple 2D manufacturing processes such as sheet cutting, sheet bending, and welding.

## **Chapter 2: Related Work**

For each of the distinct methods in the evaluation module, there is a distinct area of research. The evaluation method contains the following simulation modules: tool selection, path planning, tool wear, machining simulation, operational costing, and non-productive manufacturing times. The machining simulation module is based on well-known and published equations and fundamentals, where tool selection, part costing, tool wear, and path planning are areas of continued research. There was a surge of interest in automatic tool selection in the 1990's, most likely correlated with an interest in achieving this functionality in commercial CAM offerings. In 1991, Bala and Chang propose a method based on offset curves that automatically selected a tool for an area of complex geometry with islands [5]. This work is able to reduce the geometry to manageable polygons, and identify feasible tool access regions. Once a cutter is selected, the path for that tool can be optimized to reduce machining time. This work is unable to consider multiple tools per feature. Veeramani and Gau (1997) use Voronoi mountains to consider multiple tools per feature, where the mountains are used to analyze residual material left by the initial tool [6]. The method they present is very accurate, but their results show nearly 8 seconds of computation time to evaluate a single feature. Lim, Corney, and Clark (2000) use a proprietary feature finder to reduce parts to simple features, and used offset curves to determine tool accessibility [7][8]. They pioneer the method of allowing the feature finder to determine the smallest radius in the feature, and automatically selecting a tool based on that radius. Each feature is offset individually and the feature offsets are combined to generate tool paths and determine sizing of the "larger" tool. Carpenter and Maropoulos (2000) suggest tool selection based on several performance parameters, including surface finish, available machine parameters, and tool life [9]. All



of these bodies of work produce excellent results with regard to tool selection and path planning, but their reliance on geometric processing translates to high computational expense.

As kernel's for NC programming, they are ideal, since they simulate 3D tool sweeping, they can generate extremely accurate simulations, even providing the user with feedback on how well the generated NC code will produce the desired part. Since these simulations are driven by the very code generated, they are always accurate. The high computational expense is of little concern, since they typically must perform the simulation only once for a given part, which takes a manner of seconds. For our purposes, where simulations may need to be run  $10^3 - 10^6$  times, a single simulation requiring several seconds to complete is prohibitively slow.

A different approach to estimating cost and time for part production is the use of a database of previously manufactured parts. These previous parts contain the solid model, the feature breakdown, and the actual cost of production. Feature recognition is used to compare parts, but no tool selection or path planning is utilized. Database comparison systems include Apriori [10] and COSTIMATOR [11]. These systems can translate a solid model to a predicted cost very quickly, but do not attempt to provide the user with time requirements for given machines. The overall accuracy of these packages is yet to be proven, and unfortunately the packages were not available to us for comparison and accuracy analysis.

## **SECTION 1: MANUFACTURING SIMULATION**

This section will describe the body of work dedicated to simulating the process of manufacturing a part. Sections will be presented discussing tool selection, tool path planning, machining simulation, tool wear modeling, non-machining production times, and overall process cost estimation. Results will be presented for a number of different parts, followed by a discussion of future work and the contribution of these models to AMFA efficacy and performance.

## Chapter 3: Tool Selection

### 3.1. OFFSET CURVES

An offset curve is the set of all points that lie a fixed perpendicular distance from a given curve. For 3-axis machining, we can narrow the scope of offset curves to those that lie in a 2D plane. Examples of these curves can be seen in Figure 1 below:

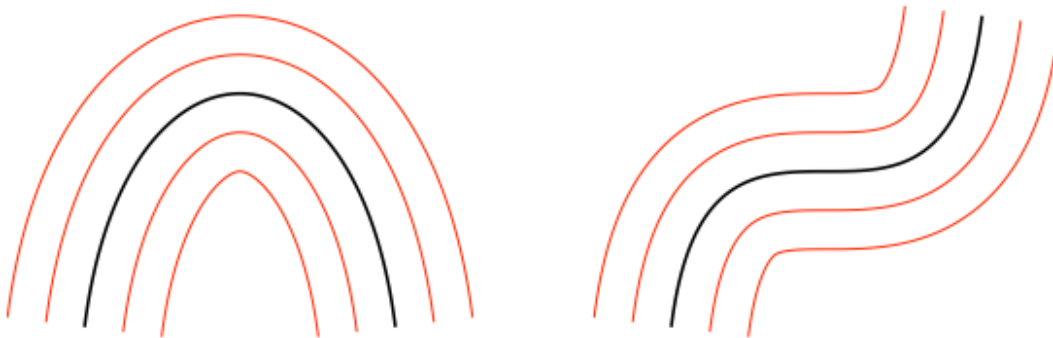


Figure 1: Examples of Offset Curves

In Figure 1 above, the original curve is shown in black, and the red curves are offsets of different distance from the original curve. Offset can be taken in either direction for any curve.

Offset curves are used today in both graphic design and CAD/CAM systems [12]. The usefulness of these curves in CAM can easily be seen, as the path required for a tool to traverse a part boundary is an offset curve of that boundary, where the offset radius is equal to the radius of the tool. For a closed loop boundary curve, offsets can be made in inwards increments of the tool radius, until no further offset curves can be generated. The combination of all of the resulting curves results in a rudimentary tool path for the

cutting of the feature enclosed by the boundary curve. This process can be seen in Figure 2 below, where the boundary curve is shown in green and the offsets can be seen in red.

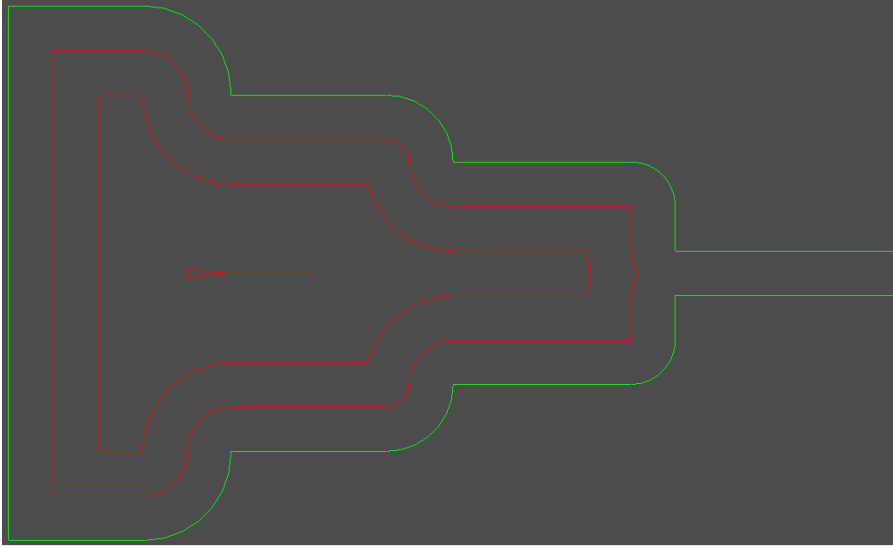


Figure 2: Offset Curves of a Specified Radius for a Given Boundary Curve

The offset curves used by AMFA are generated using the open-source package Open CASCADE, or OCC [13]. For any given partition, OCC libraries will be used to generate offset curves inward from the boundary curve of each planar face in the partition, with an increment of 0.5 mm.

The way in which AMFA utilizes offset curves of boundary curves to select tooling and perform tool path planning is unique, and will be presented in Chapters 3-4.

### 3.2. TOOL TYPE SELECTION

For a milling operation, the tooling available can be broken down into two broad categories: cylindrical cutters and spherical cutters.

Cylindrical cutters are the most common type of end mill, and are used whenever possible, due to their superior material removal rate, and lower cutting forces [14]. All cutters whose cutting profile is best described as a cylinder fall into this category. AMFA considers only the most common type of cylindrical cutter, the end mill. Selecting a different type of cylindrical cutter is very difficult even for machinists, and the ability to reason about tool type description on this level has not been achieved with the software. A few examples of end mill cutters can be seen below in figure 3.



Figures 3: Typical Tools Considered for use by AMFA

Spherical cutters are only used when required by the cutting geometry, as they take much longer to machine a given part because of their dramatically lower material removal rate. The most common type of spherical cutter, and the only type considered by

AMFA, is the ball mill. The ball mill has a hemispherical cutting tip, and a cylindrical body. A typical ball mill can be seen below in figure 4.



Figure 4: Typical Ball End Mills Considered for use by AMFA

Since the representation schema of AMFA selects the type of tool required, it will always pick an end mill, unless there is a curved surface to be machined in an orientation such that an end mill cannot machine it. In figure 5.a, the selected tool type is an end mill, since the fillet is in the direction of cutting, and there are no fillets on the face that the top of the mill machines. In figure 5.b, however, the presence of a fillet on the face machined by the tip of the end mill requires a ball mill to be selected for machining.

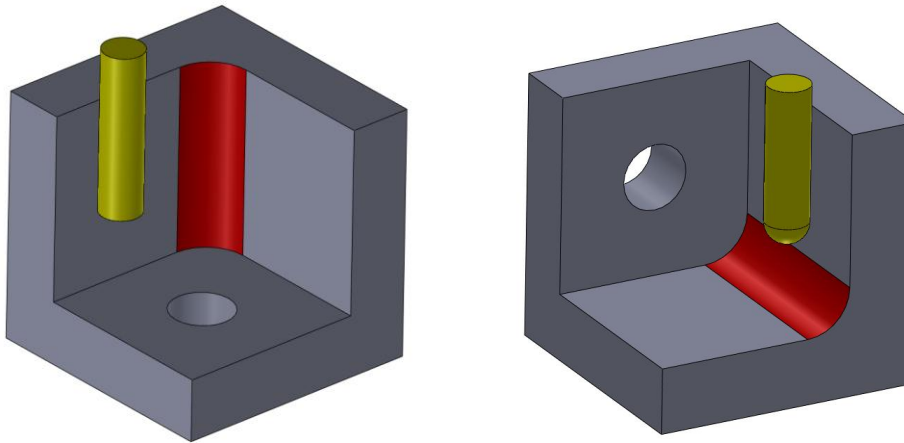


Figure 5.a,b : Partitions that Require End Mills and Ball Mills

### 3.3. OFFSET STORAGE

This work presents a unique algorithm for using offset curves to determine tool size and accessibility. Unlike related work which uses the geometry of the curves to perform 3D sweeping or other computationally-intensive tasks to determine appropriate tool sizing and accessibility, this work considers only the length and offset increment of each curve. This naturally results in a much lighter, faster performing algorithm, as the software deals only with simple mathematical calculations on an array of doubles known as ShapeData. There are also clear drawbacks to operating without knowledge of the geometry of the part, which will be discussed in the following section.

For a given partition, there is a starting face selected by representation. Provided this starting face is a closed-loop face, offset curves are generated by OCC to the inside at a standard increment, until no further curves can be generated. The length of each of these curves is stored to the array of doubles ShapeData in a specific format, shown in Table 1 below:

array index	
0	smallest corner radius
1	increment of offset curves
2	length of boundary curve
3	length of curve offset 1 increment
4	length of curve offset 2 increments
.	.
.	.
m-1	length of curve offset n-1 increments
m	length of curve offset n increments

Table 1: Format of the Shapedata Array

An example of a starting face boundary curve, its associated offset curve, and the resulting shapedata array can be seen in figure 6 below. The pocket at the top of Figure 6 is a feasible boundary curve, with offset curves generated until no further offsets are possible at the increment chosen. What is stored in the partition is the array shown at the bottom of the figure. ShapeData[0] always contains the smallest radius in the boundary curve. If there is no radius in the boundary, this cell will contain 0.0. ShapeData[1] contains the chosen increment for consecutive offsets, in this case, 0.5 mm. ShapeData[2] is the length or perimeter of the boundary curve. From ShapeData[3] until the last entry, the numbers represent the length of the offset curves. Each position represents another increment from the previous number. For Example, ShapeData [3] = 0.5 mm from boundary, ShapeData[4] = 1.0mm from boundary, etc. A summary of the format of ShapeData can be seen below in figure 6.



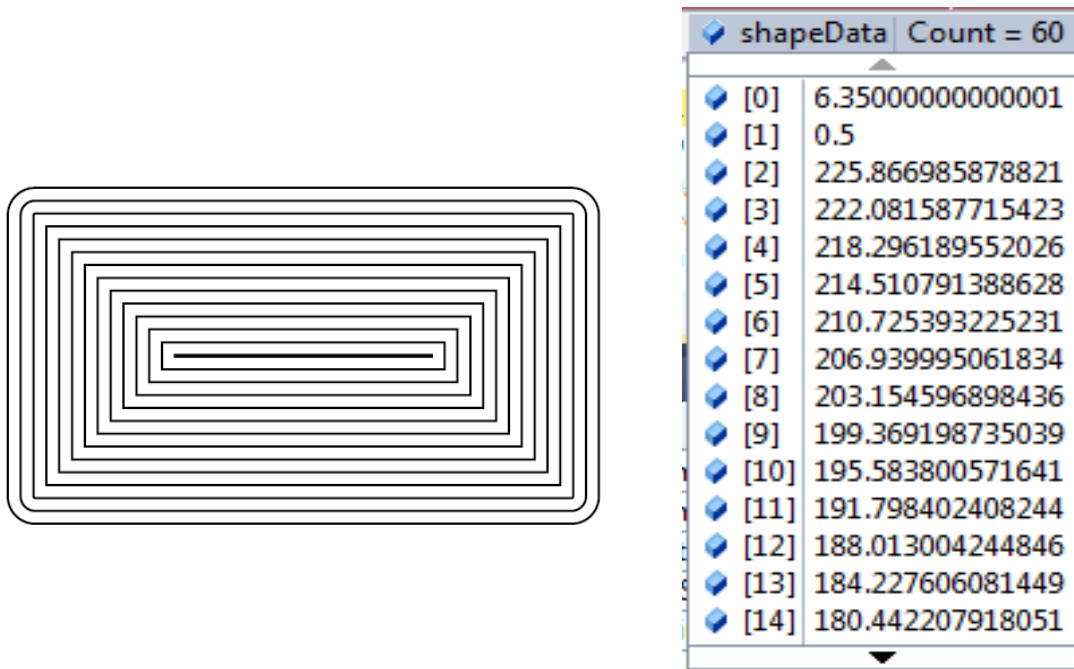


Figure 6: Conversion of Offset Curves to Shapedata Array

### 3.4. STARTING FACE ASSUMPTIONS

#### 3.4.1. Geometric Requirements

Due to the nature of the decomposition algorithm employed by AMFA, there are geometric constraints placed on the starting face that are greatly useful for tool selection and path planning. The most important of these constraints is that the starting face will always be convex. This means that a tool can be sized based on the geometry of the boundary alone, without concern of leaving “islands” of material, or hitting “pinch points”, where the face becomes too small to pass, leaving material on the other side.

### **3.4.2. Virtual boundary assumptions**

The nature of convex decomposition will result in many partition faces with sharp edges. Since these partitions are evaluated individually, a tool path must be planned on this face alone. A start face with sharp corners is technically a non-machineable face, since the cylindrical tool cannot access the sharp corner of the face. However, each time this occurs for these partitions, it is because the partition is either: mated to another partition at the sharp corner, or, is external and accessible from the outside of the part. Therefore, any sharp internal corners in the partitions processed by the evaluation module are really “virtual” corners, and can be ignored. This assumption will lead to small errors in the path planning for tooling, which will be addressed in Chapter 4. A starting face containing two “virtual” sharp corners can be seen below in figure 7:

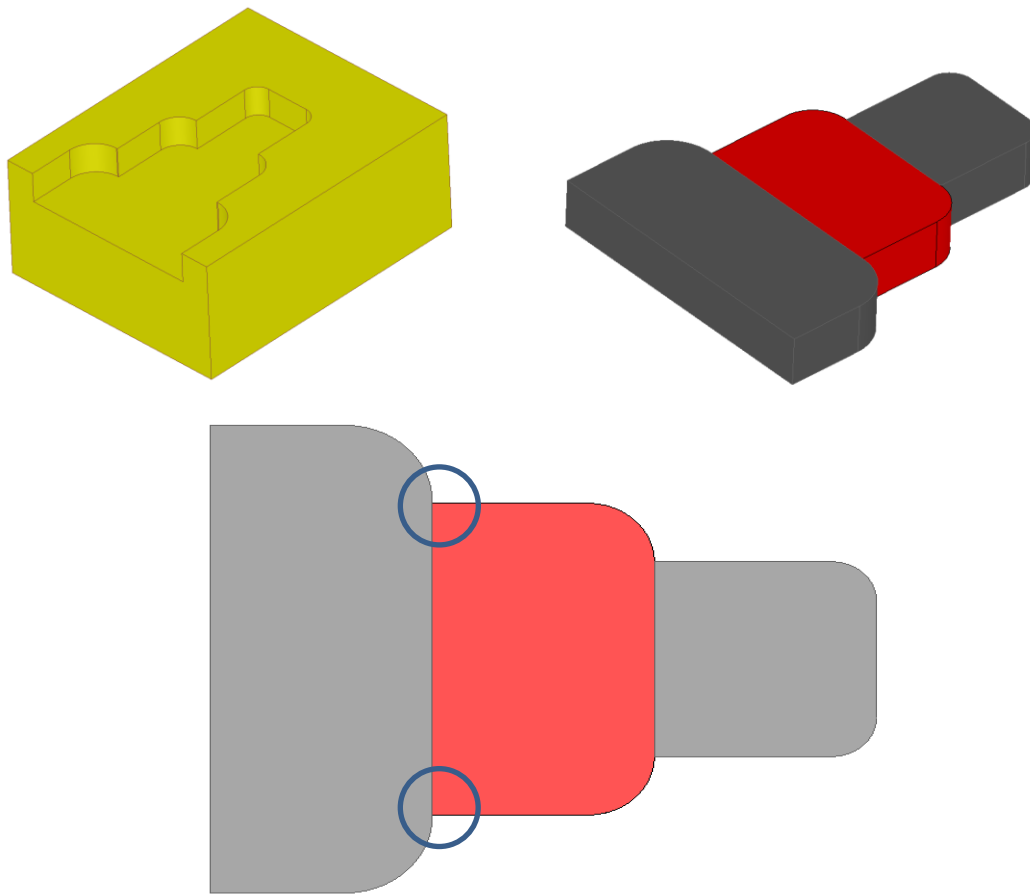


Figure 7: Virtual Sharp Corners on a Partition Mated with Another Partition

### 3.5. TOOL SIZE SELECTION

The specific format of the information stored reveals information about the original shape. The first number in the array is always the radius of the smallest curvature in the path. This allows for rapid assessment of what the smallest tool required to machine the area is, since the tool must have a radius smaller than the smallest radius of curvature on the part in order to not leave any residuals.

The length of the offset curve array allows for the max tool size for accessibility to be established. The number of offset curves present and the increment of each offset determine the radius of the largest tool that will fit in the area.

$$R_{tool} := N_{\# \text{ of offsets}} * R_{offset} + R_{offset} \quad (1)$$

The geometric equivalent of this equation (1) is shown in Figure 2.

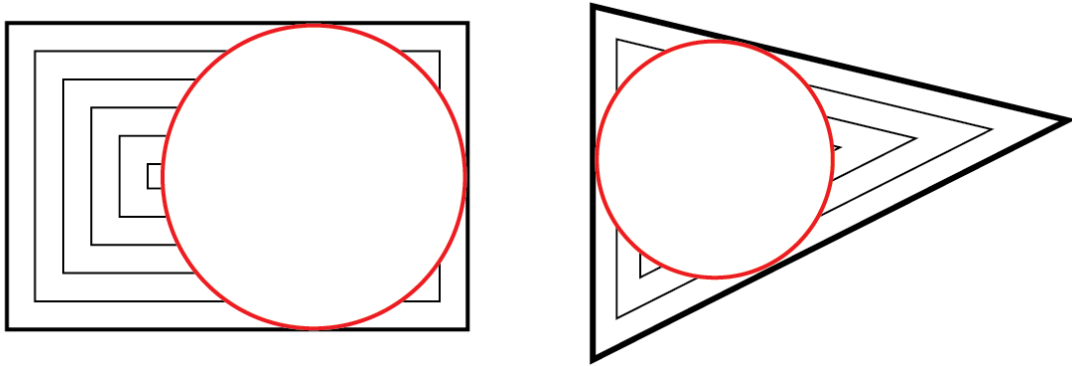


Figure 8: Maximum Tool Diameter for Accessibility

Equation (1) is true even for non-square or rounded shapes, although the area accessibility will change for each of these scenarios (notice the large amount of

inaccessible area to a tool of this size for the triangular geometry, Figure 8.b). Although this gives the largest tool that is possible to fit into an area, it is neither particularly useful nor realistic to select a tool of this size for machining. We assume for any real machining scenario that the tool chosen is closer to half of the smallest internal dimension, allowing for a very similar tool path length, but more material removal and less machine loading. Therefore, we take the equation given in equation 1 and choose a tool with a diameter of this value instead of the radius, thus the ideal tool is half the size of the one shown above. It is shown later that this tool selection closely matches the tool size choices by both machinists and automatic selection by CAM software.

$$D_{tool} := N_{\# \text{ of offsets}} * R_{offset} + R_{offset} \quad (2)$$

The geometric equivalent of this equation (2) is shown in Figure 9.

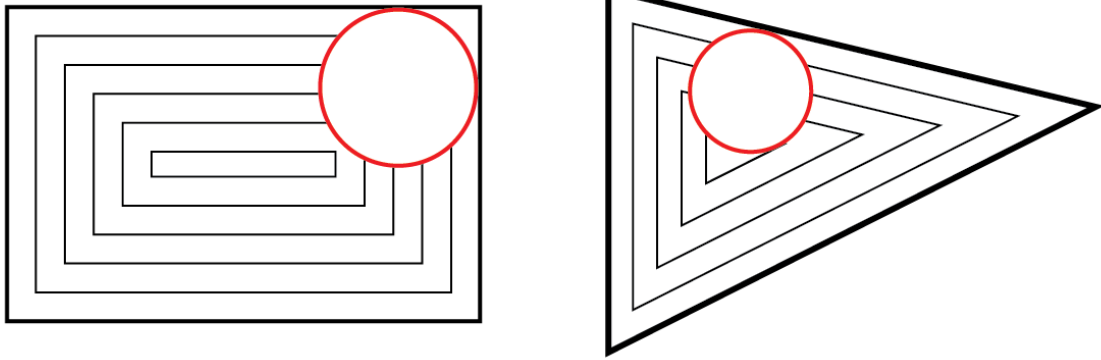


Figure 9: Chosen Tool Diameter for Machining

The extra  $R_{offset}$  term in equation 2 is to ensure that the diameter of the tool chosen exceeds half the size of the feature, so that if it traverses the boundary curve, no

island of material is left in the middle. It is appropriate because the final offset will not reach the center, due to the finite increment value. The distance from the final offset to the middle could be anywhere between 0 and 1 increment, so adding an increment to the diameter ensures that the middle is reached. Should the machinist prefer to use smaller tools, this is easily adaptable to selecting a tool that is any size less than that given by equation 2.

In some situations, this large tool is all that is required to be used. This case exists when the smallest radius in the boundary (given in ShapeData[0]) is smaller than the tool diameter, given by equation 2, or is zero. For the two shapes given in Figure 3 above, this tool would be the only tool required to machine the partition, since there are no radii in the boundary. This means that the tool has 100% area accessibility, since sharp corners represent “open” boundaries, as discussed in Chapter 4.

If, however, the shape has a corner with a small radius, the part has area that is inaccessible by this tool, which necessitates the use of a secondary, smaller tool. The size of this smaller tool is easy to select, since the first entry in the offset curve data gives the radius of the smallest radius corner.

Although the size of each of the tools is easy to compute, we need a check to determine if two tools will be required for the partition. This check is simply a check between the diameter of the large tool and the diameter of the small tool, and is computed as follows:

$$R_{bigtool} - R_{fillet} \geq X \text{ (mm)} \quad (3)$$

If equation 3 is true, two tools will be selected. The parameter (X) is a user-defined value. The value of X is taken to be a value greater than zero, and the actual value of X should be found through experimentation to ensure the lowest machining time

for the part. The reason  $X$  is typically greater than zero is because of the time penalty for using multiple tools for a given machining operation. In order to switch tools, the machine must return to the datum, physically switch the tools, return to the part, and begin machining again. The time required to perform these operations can be greater than the time penalty of using a slightly smaller tool to machine the entire partition. When the size of the tool has been decreased enough that the machining time is now greater than the machining time of the larger tool plus the total tool change time, using two tools to machine the partition will result in a total time savings. Thus, the value of  $X$  would ideally be computed on a per-part basis using an optimization routine. For this work, and to further ease the computational requirement of the calculation, a set value of  $X = 5\text{mm}$  is used, as this value has been found to be near-optimal for most cases.

In addition to calculating the ideal tool size(s) for machining the feature it will also find actual tools available to the machinist. Using a database of available tooling, the software will load tool data for a tool that most closely matches the parameters calculated here. For the large tool we are looking for a tool that has the closest diameter to that given by equation 2, with the requirement that the actual tool diameter be larger than equation 2. For the small tool, if required, we are looking for a tool that most closely matches  $\text{ShapeData}[0]$ , the smallest radius in the partition, with the requirement that the actual radius be smaller than the desired radius. Therefore we scan the database and load whichever tool satisfies the criterion:

$$\begin{aligned} & \min\{ \mathbf{BigD}_{actual} - \mathbf{BigD}_{calculated} \} \\ \text{s.t. } & \mathbf{BigD}_{actual} - \mathbf{BigD}_{calculated} \geq 0 \end{aligned} \quad (4)$$

$$\begin{aligned} & \min\{ |\mathbf{SmallD}_{actual} - \mathbf{SmallD}_{calculated}| \} \\ \text{s.t. } & \mathbf{SmallD}_{actual} - \mathbf{SmallD}_{calculated} \leq 0 \end{aligned} \quad (5)$$

One common problem is when the calculated size for the large tool is larger than any tool in the database. The database is limited (currently) to 1” diameter, and it is common for partitions to be larger than this. In this case the largest tool available will be chosen.

Once the most appropriate tool in the database is found, several of the parameters for that tool are loaded into the software, for later use by other modules. The data that must be loaded for each tool is presented in table 2:

geometric parameters	diameter (inches)
	exposed length (inches)
tool design parameters	material
	number of flutes
tool wear constants	C
	X
	Y
	Z

Table 2: Relevant Tooling Parameters

### 3.6. LIMITATIONS AND FUTURE WORK

Although tooling is currently selected based only on diameter, future efforts in AMFA will focus on the selection of tooling based on additional parameters, such as tool material, number of flutes, tool coatings, etc. The main difficulty is implementing selection processes on these parameters is that selection of tooling in industry is, in general, based on “rules of thumb” rather than hard constraints. Although there are instances where the part material disqualifies certain tool materials, typically tools of different materials, coatings, or number of flutes can all be used to machine a given part.



Further advances in the simulation, such as the ability to reason about surface finish and tolerance, will allow the selection of these parameters to become more concrete and verifiable.

## Chapter 4: Tool Path Planning

Once tools have been chosen for a partition, in order to estimate the time to machine the part, we need to determine the amount of time required for the tool to remove the partition. This estimate will be based on the linear feed rate of the machine, calculated in the machining model in Section 4, and the path length of the tool. Once again, the offset curve data stored in ShapeData will allow us to estimate the path length required of the tool(s). Initially we will assume the entire area will be machined with a single, or primary tool. If needed, we will consider the use of a smaller tool.

### 4.1. PRIMARY TOOL

Assuming we have selected a large tool using equation 2, we know that the tool is slightly more than half the internal dimension of the partition. This is significant because it means with one sweep of the tool tangent to the boundary curve, the tool will remove all material accessible to it. The curve that is offset from the boundary an amount equal to the radius of the tool is the centerline along which the tool would travel to achieve this. Since the offsets are in increments, we have to find the offset increment that most closely matches the radius of the selected tool. It is not important if we over or under-estimate the size of the tool, as the maximum error is half of an increment (0.25mm in this case). An example of a selected offset curve for tool travel can be seen in Figure 10.

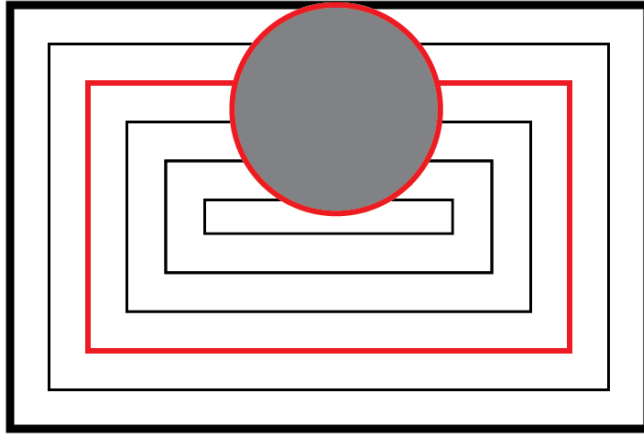
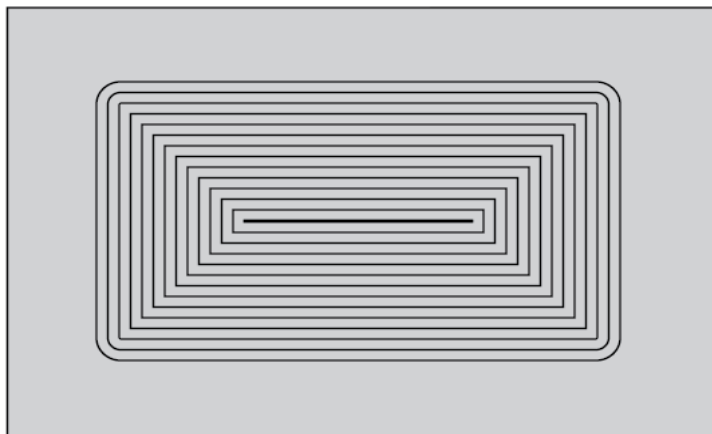


Figure 10: Boundary Tangent Tool Path

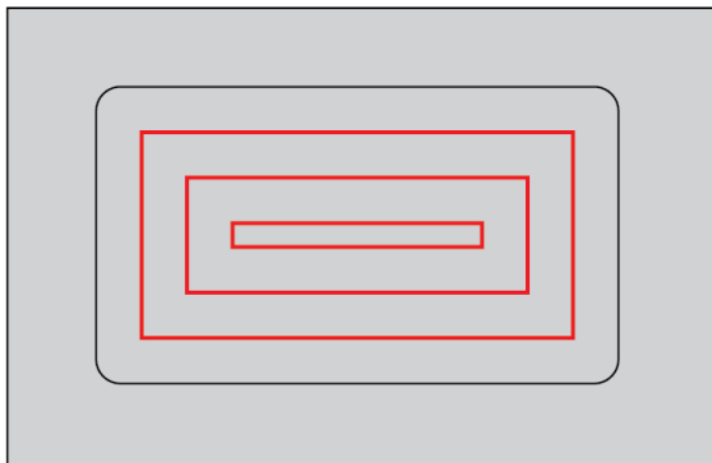
If the tool in Figure 10 is swept along the curve highlighted in red, it will remove all of the material except the corners. This corresponds to a 100% embedded tool, as the entire diameter of the tool is cutting during almost the entire motion along the path. Situations like this are very uncommon in machining, and commercial CAM package FeatureCAM [15] approaches path generation in a different manner. To replicate a more realistic cutting approach, a second curve is selected, this time at twice the radius of the tool, corresponding to the final (or nearly final) offset curve stored in ShapeData. This second curve will be traversed first, making a “hole” in the center, and then the second curve will be traversed, in this case with a 50% step-over. The combined length of both of these curves gives the path length of the tool to machine the pocket. If a smaller tool is used, the offset curves are selected such that their offset from the boundary is multiples of the tool radius. An implementation of this offset curve selection method and comparison to a FeatureCAM cutting path for the same shape can be seen in Figure 11. Figure 11(a) shows an internal pocket with offset curves stored in ShapeData. Using

a tool that is 25% max tool size for accessibility, offset curves selected for tool travel have been highlighted in red in Figure 11(b). In Figure 11(c), a FeatureCAM simulation for the same model is shown, where the productive portion of the tool path can be seen in black. It is clear that the tool paths in figure 11.b and figure 11.c are nearly identical.

A)



B)



C)

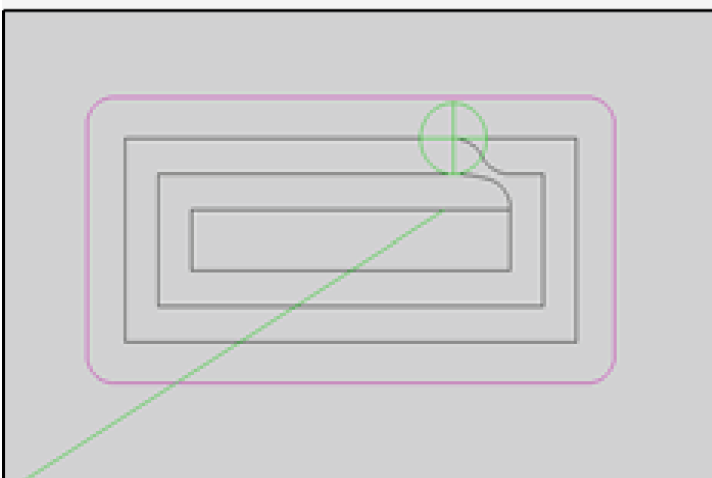


Figure 11: AMFA Toolpath Compared to Popular CAM Package Toolpath

One difference that can be observed between our tool path curves and the tool path created by FeatureCAM is the tool motion between curves. We need to account for tool motion to jump from an inner curve to an outer one. As the tool path in figure 5.c shows, this travel is more complicated than the distance between the curves. While FeatureCAM uses a variety of curves to move between cutting curves, we assume that the jump is done at a fixed angle from the curve. After some trial and error, an angle of 20 degrees has been found to most closely replicate the distance travelled by FeatureCAM, although this is an easily adjustable parameter in the model. The additional path length is shown in red in figure 12.

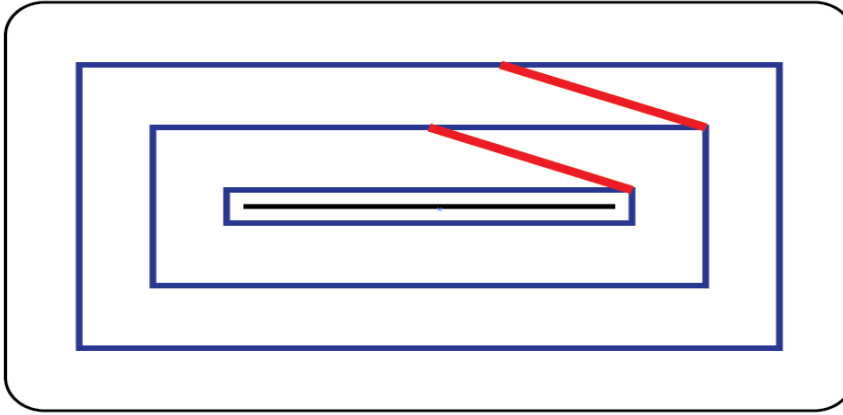


Figure 12: Additional Path Length to Travel Between Offsets

The additional path length shown in Figure 6 above is in general calculated with the following equation:

$$\textit{Join Offsets Distance} := \frac{R_{tool}}{\sin(20 \text{ degrees})} * (N_{offsets} - 1) \quad (6)$$

Where  $N$  is the number of selected offsets (3 in figure 12). Again, the use of the tool radius as the distance between curves corresponds to a 50% step-over value. At this point we have an estimate of the planar path length for a given tool. This method applies only to the primary (large) tool, and will be carried out exactly the same way regardless of whether a secondary tool is required. Adjustments to the planar path length are necessary to consider multiple passes, depending on the depth of the partition, which will be addressed later in this chapter.

#### **4.2. SMALL TOOL**

The primary complication of using a smaller secondary tool is being able to simulate that we do not need to recut everything the primary tool was able to access. Since we have no geometric information about the shape or even the offset curves used as a tool path for the large tool, the approach we take is to estimate the planar area accessed by the larger tool, and take the difference between this area and the face area as the amount of material that the small tool is required to remove. This requires the determination of the area removed by the large tool.

Estimating the area not machined by the large tool is a non-trivial calculation, and in general, without using the geometric information about the area, can be estimated only. More intensive methods, such as 3D sweeping, or even Minkowski differences, can generate extremely accurate values for area machined by a tool of a given size. Both of these approaches, however, have drawbacks that negate the benefits of their improved accuracy.

3D sweeping is a powerful tool that results in known and exact information about material removed and left behind by a tool passing through material. Many CAM systems use this tool, as it is generally the most accurate method available [16]. The

drawback is that it is very computationally intensive [17]. Given the branching nature of the AMFA search, and the number of times this area must be estimated, using 3D sweeping would slow down the computation an unacceptable amount.

Minkowski sums are able to generate the area accessed by moving one shape inside another. However, there are specific limitations on the kind of shapes that may be used. The most important of which is Minkowski sums require a set of discrete points, and assumes these points are connected with straight lines. This makes computing the difference between a circular-face tool, and a boundary containing radii curves, very difficult. To perform the calculation with accuracy would require the interpolation of these curves to a large number of points. The computational expense of the Minkowski approach is low for boundaries containing only straight lines, but due to the necessity to interpolate curves, the computational demand grows large when the boundaries contain fillets (radius in part) or circles (cutting tool).

As it turns out, we are able to estimate the area removal by the large tool using only the data contained in the ShapeData array, resulting in a very light computation, and the area of this method is computable for comparison with alternative methods.

#### **4.2.1 Area removed by big tool**

Given the offsets selected to be part of the tool path for the big tool, we can determine the area accessed by this tool. Fundamentally, if a tool is swept along a straight line, the area covered by the tool is known. If we assume that the tool is run with a typical stepover value of 50%, the area covered by the tool can be given by the radius of the tool and the length travelled, as shown in figure 13 below:



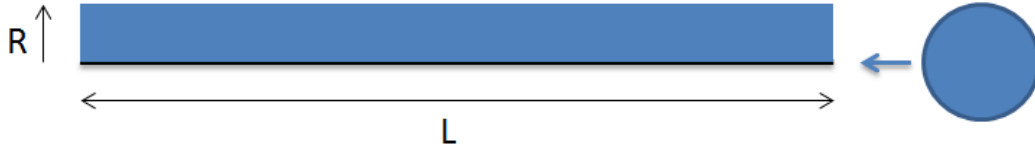


Figure 13: Area Covered by Sweeping a Tool Along a Line

Thus the area removed by sweeping a tool at a given stepover can be given by the following equation.

$$\text{Area removed by tool} = L * D * \text{Stepover}_{\%} \quad (7)$$

Where D is the diameter of the tool, and Stepover is the percentage of the tool that is embedded in the part. Since the tool path selection operates with a value of the stepover at 50%, the Stepover term can be removed, and the diameter of the tool replaced with the radius of the tool. The area removed can now be computed using the following equation:

$$\text{Area removed by tool} = L * R \quad (8)$$

While equation 8 gives the area removed sweeping the tool along a straight line, we are more interested in the effect of sweeping the tool along one of the offset curves. This reduces to calculating the area between any two offset curves separated by distance R. If the outermost curve length is given as L, the area between these two curves is given as:

$$\text{Area between curves} := L * R - X * R^2 \quad (9)$$

Where  $X$  is a coefficient that depends on the number of sides of the curve, amount of boundary that is curved, and overall shape of curve (variance in internal angles). Since none of these values are known, the value of  $X$  must be estimated. The second term in equation 9 shall be called the corner adjustment term, and it is the only source of error in this estimation. Note that the error depends only on  $R$ , so as  $L$  becomes large with respect to  $R$ , the relative error becomes very small.

The inclusion of the corner adjustment term is necessary, because using the straight line area removal will include errors, since the curve is a closed loop. The corners will include areas that are “overlapped”, or considered twice, making the area estimation too large. For a rectangle, the exact value of  $X$  can be easily deduced, as seen in the following figure:

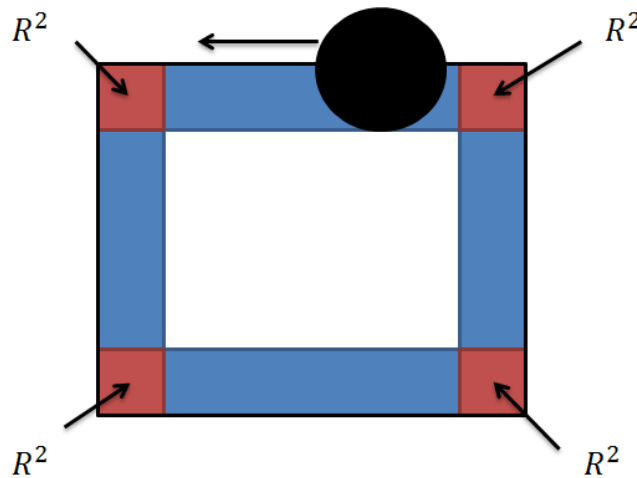


Figure 14. Areas of Overlap for Tracing a Square Boundary

In figure 15 above, the areas shown in red are overlapping areas, and to correctly determine the area accessed by the tool, the combined total of the areas shown in red must be removed. This area is  $4 \cdot R^2$ , making the value of  $X$  to be 4. In this example, the value of area removed can be determined exactly, because the number of sides and

the interior angles of the boundary loop are known. To determine a standard value of  $X$  to be used for all calculations, we must understand the effect of changing the number of sides and the interior angles of the boundary.

To understand the effect of the number of sides on the value  $X$ , we begin by looking only at regular polygons with all internal angles equal. Using trigonometry to quantify the area of overlap of each corner, the value of  $X$  can be determined using the following equation:

$$N \left( \left( L * \frac{r}{N} - 2r^2 \tan \left( \frac{\pi}{2} - \frac{(N-2)(\frac{\pi}{2})}{N} \right) \right) + r^2 \tan \left( \frac{\pi}{2} - \frac{(N-2)(\frac{\pi}{2})}{N} \right) \right) \quad (10)$$

Using equation 10 given above, we can plot the resulting value of  $X$  vs. the number of sides for regular polygons, as shown in figure 15 below.

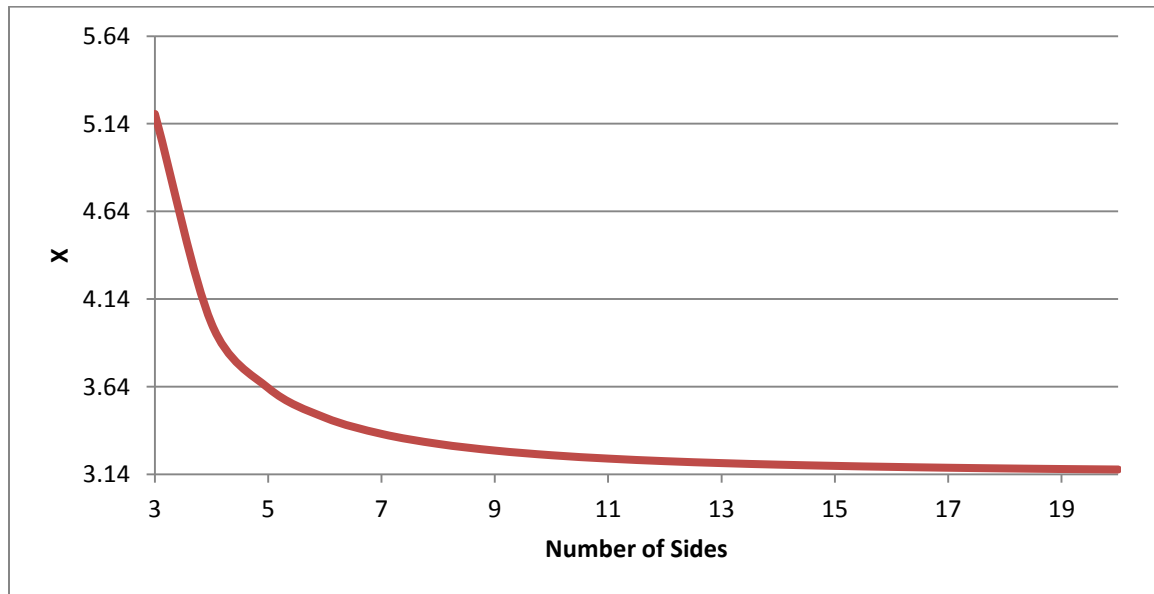


Figure 15: X Versus the Number of Sides for Regular Polygons

It can be shown that as the number of sides goes to infinity, the value of  $X$  converges to  $\pi$ .

$$\lim_{n \rightarrow \infty} N \left( \left( L * \frac{r}{N} - 2r^2 \tan \left( \frac{\pi}{2} - \frac{(N-2)(\frac{\pi}{2})}{N} \right) \right) + r^2 \tan \left( \frac{\pi}{2} - \frac{(N-2)(\frac{\pi}{2})}{N} \right) \right) = rL - \pi r^2 \quad (11)$$

This is logical, as one would expect the adjust term for a circle (infinite sided polygon) to be  $\pi$ .

Regular polygons are uncommon in partitions created by AMFA. While analyzing regular polygons has offered valuable insight into the effect of the number of sides on the overall error of our calculation, we need to further analyze non-regular polygons. It is clear from figure 15 that a 3-sided object has the largest possible errors, so we choose to investigate the effect of changing internal angles for a 3-sided polygon. Quantifying this worst-case scenario allows us to firmly claim a maximum deviation on any calculation we run, and report this as a relative error. Since the sharpest corners have the highest error values, the worst case shape will have two such angles. To keep the analysis reasonable, we must limit how small these angles can be, since the error will propagate to infinity as the angles approach 0. To keep the area of the triangle a reasonable shape, the corners are limited to a minimum of 20 degrees. To understand the effect of the shape of the triangle on the total area, the two equal angles are incremented identically, until the third angle is only 20 degrees. The results of this analysis can be seen below in Figure 16:

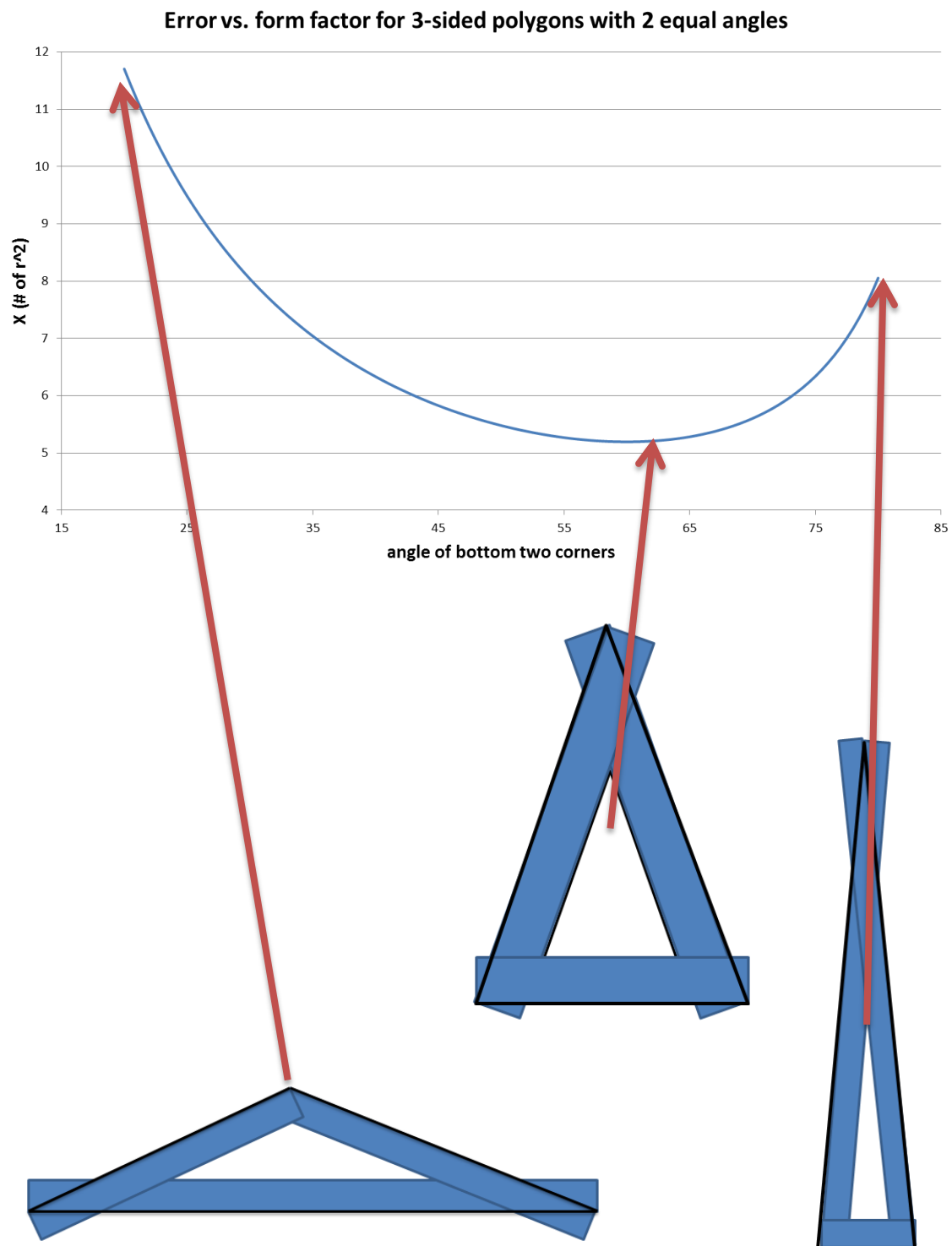


Figure 16: Effect of Changing the Shape of a Triangle on Error

From figure 16 we can see that the error for a triangle can range from  $5.15(R^2)$  to  $11.7(R^2)$ . We know that for a 4-sided shape, the error can range from 4 (perfect square) to the same  $11.7(R^2)$ . The upper bound on the error is the same as with the triangle, because it is always possible for a higher-sided shape to take the shape of a lesser-sided shape, therefore an N sided figure could have an error as large as  $11.7(R^2)$ . However, these cases are rare, and by looking at several typical shapes processed by AMFA, an X value of 6 has been found to result in the lowest typical error. Using this value, a study on a range of machined parts was done and the error in the area removal was found to be typically less than 5%.

To determine the area, we start at the innermost curve and move outwards, so we will add the corner adjustment term rather than subtract it. We will also make an adjustment for the fact that on the final pass, the exact area of removal is known, since the radius of the tool is known.

$$\text{Area removed by large tool} := \sum L_{\text{toolpath}} * R_{\text{tool}} + X * (N_{\text{offsets}} - 1) * R^2 + \pi * R^2 \quad (12)$$

Where L is the length of each offset selected to be part of the toolpath, and R is the radius of the tool. The difference in equation (12) and the area of the face (known) will give an estimate of the area of the face inaccessible to the large tool, which the small tool will have to remove.

Translating this value to a path length requires another trick, using the step-over value for the tool. Given the diameter of the small tool, and the area the tool must remove, dividing these values gives the length the tool would have to travel to cover the

area. This does not account for unproductive tool travel, which can be a significant portion of tool travel for a “cleanup” tool. For example, when the tool must move between unconnected sections of residual material. The assumption is made that since these shapes are convex, the area left to be machined is in the corners of the shape, so the length of the boundary curve is added to the tool path for the small tool, to account for the fact that the small tool must travel to all of the corners to machine them. Therefore the small tool path length can be given as:

$$\text{small tool path length} := \frac{\text{Area of Face} - \text{Big tool Area removed}}{\text{small tool diameter} * (\text{step-over})} + \text{finishing distance} \quad (13)$$

Where “finishing distance” is an adjustment to account for the non-productive segments of tool travel that must be performed to access the material to be machined. The computation of this value depends on if the partition is internal, or has faces in common with the bounding box (open to the air).

#### 4.2.2 Finishing Distance

Estimating the portion of travel of the small tool that is non-productive is essentially measuring the distance of a finishing pass. For fully interior pockets, the boundary curve may be used as the non-productive tool path distance, since to access all four corners of the pocket, the tool would have to travel the entire boundary curve. This method breaks down, however, when machining partitions that are either open to the air, or adjacent to other partitions that require machining. For these circumstances, using the boundary curve as the non-productive tool travel distance will overestimate the required

travel. For a pocket with two faces open, this distance will be nearly twice the required distance.

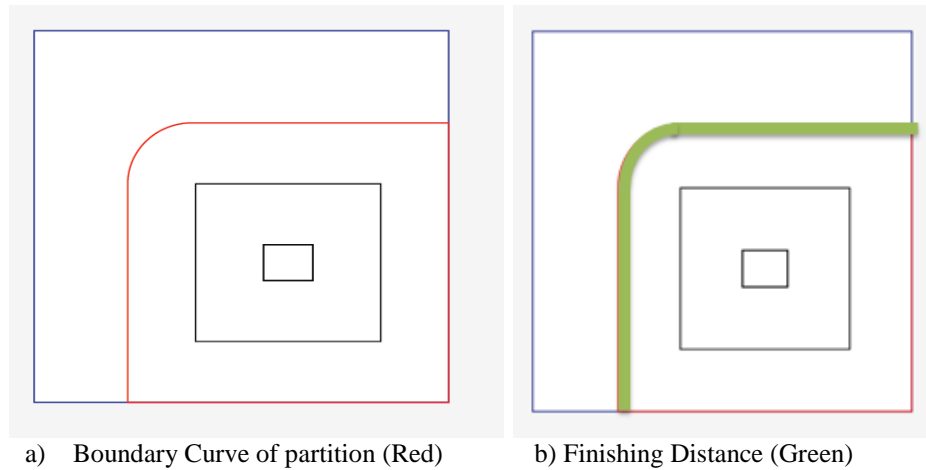


Figure 16: Boundary Curve vs. Finishing Distance for Open Pocket

In figure 16.a, the red curve shown is the boundary curve of this partition. The curves inside of this are the tool path offsets. Figure 16.b shows the portion of the boundary curve that requires finishing, highlighted in green. We need to be able to trace the boundary curve of partition and determine how much of the distance is required for machining. As stated previously, this is analogous to the distance required to get the tool to where it removes the leftover material, which is the distance we need.

A boundary curve is represented in the graph as a set of nodes connected by edges, either straight or radial. Each of these edges contains labels that will allow us to determine whether this edge needs to be considered in the calculation of the finishing pass length. The only edges that need to be considered are edges that are shared between the partition being removed, and the final desired part. The labels in the edges allow us to determine, for each edge, if the edge is shared with the part. If so, the edge length is calculated and added to the finishing pass length. For example, consider the solid model



shown in figure 17 below. The negative (parts that must be removed during machining ) of this part appears in figure 18.

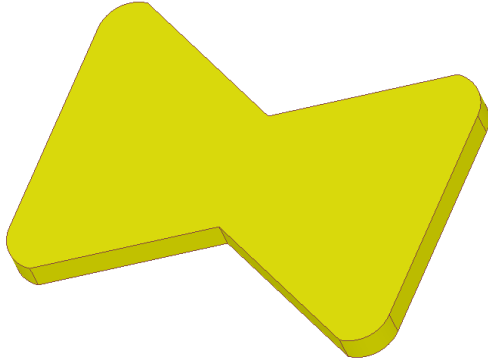


Figure 17: Model to Demonstrate Finishing Length Calculation

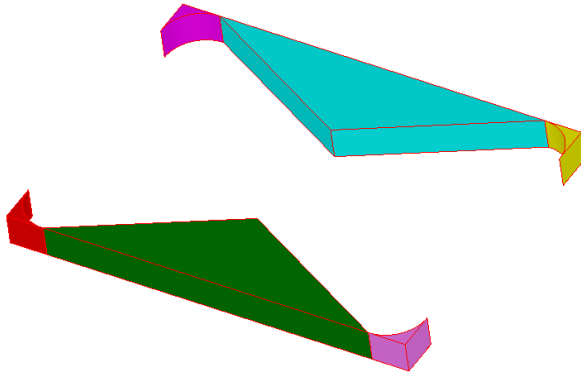


Figure 18: Negative of Solid Model Shown in Figure 17

The negative shown in figure 18 is made up of 6 partitions, each represented with a different color. Since AMFA processes models one partition at a time, for each of the 6 partitions, we need to evaluate which edges require finishing and which do not. Using the edges and their associated labels, the edges can be identified, as shown in figure 19.

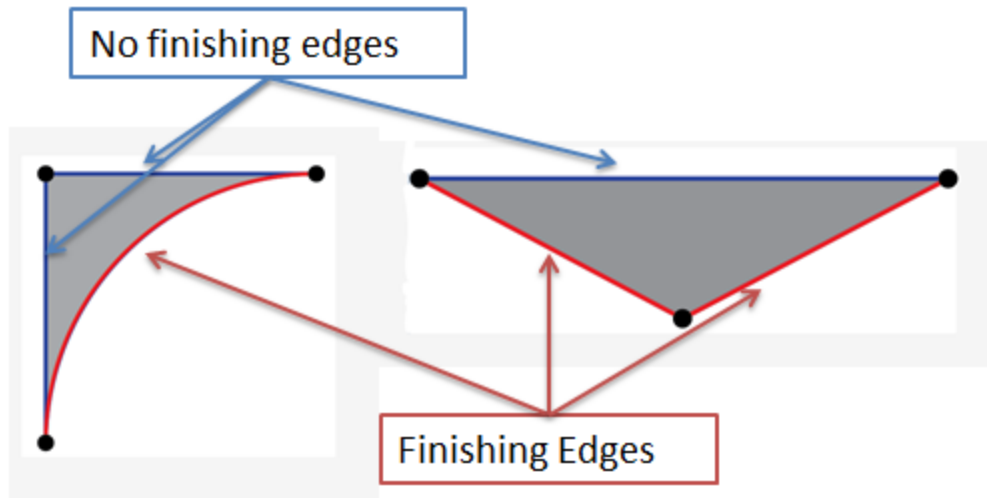


Figure 19: Identifying Edges for Finishing Pass

The lengths of all the finishing edges in a given partition are calculated and summed, and recorded as the non-productive path length and added to the total path length for the small tool.

#### 4.3. TOTAL TOOL TRAVEL

Similar to the large tool case, equation (9) gives the planar path length of the small tool. However, most cutting circumstances require multiple depth steps, or “z-steps” to reach the required depth. Since each cutting simulation will have a prescribed Depth of Cut (DOC), we can determine if, and how many, z-steps are required. The complication is that the depth of the partition is unknown. The only information available is the volume of the partition and the area of the face.

Using the volume of the partition and the face area, we can estimate the depth of the partition. Immediately, the concern is raised that this is only accurate for prismatic partitions. While this is true, for non-prismatic shapes, the incorrect calculation of the depth will still lead to a good estimation of total tool travel required. The tool travel is essentially an estimate of the total amount of area that the tool must cover. The total area is the area of the partition at the depth specified by the depth of cut of the tool and the number of z-steps that have been cut. Given any partition and starting face, the part is treated in the tool path planning stage as a prism. Take for example the three parts shown in figure 20:

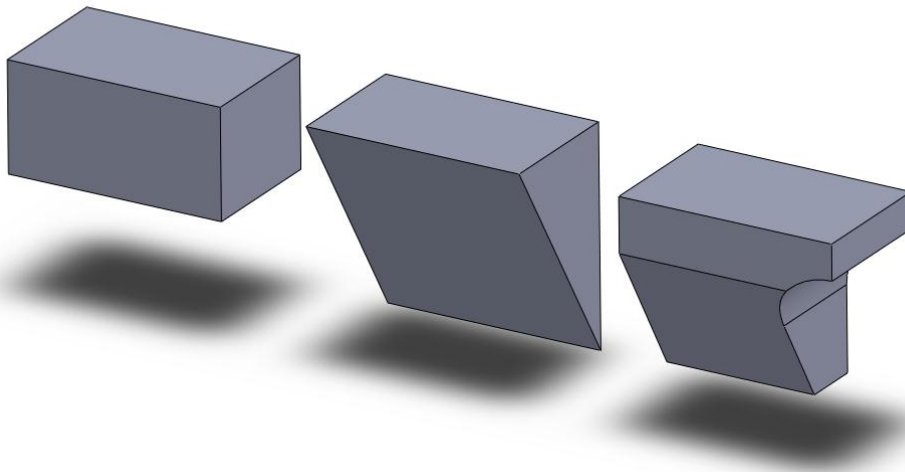


Figure 20: Various Parts of Equal Volume and (Top) Surface Area

Each of the parts shown in figure 20 have identical volume, and the top faces all have identical area. The tool path planning method would reduce all of these parts to the prism shown in figure 20.a, and calculate the prismatic depth, and the number of z-steps required, where each pass is the same area. It is clear from the non-prismatic shapes in 20.b and 20.c that each z-step will result in a different area than the top surface. It can be

shown, however, that the total area covered by the tool is identical for all cases. For the shape shown in figure 20.b, the area of the part decreases as the tool progresses in the z-direction. There are, however, many more z-steps required since the part is much deeper than its' prismatic counterpart. It is clear that as the DOC (z-distance between machining planes) becomes infinitesimally small, the total area of all cut planes essentially measures the volume of the part. Since all parts have the same volume, all will result in the same cutting area. Therefore we can conclude that specific knowledge of the shape of the partition is unnecessary for this element of tool path planning, and would only add additional computational expense. Using a finite-valued DOC will result in small errors in the calculation, which is accepted as inevitability.

Given depth, we can determine how many Z-steps are required, given the DOC determined by the tool diameter. These calculations are performed as follows:

$$\mathbf{partition\ depth} := \frac{\mathbf{Volume}}{\mathbf{Face\ Area}} \quad (14)$$

$$\mathbf{N_{z\ steps}} := \mathbf{Ceiling} \left( \frac{\mathbf{partition\ depth}}{\mathbf{DOC}} \right) \quad (15)$$

The ceiling function is required since the depth is likely to not be a perfect multiple of the DOC, yet we need an integer value for number of z-steps. The result is that the final pass will have a smaller DOC than all the previous passes, which reflects the approach CAM software takes with multiple z-steps.

Similar to the adjustment taken to add path length to account for the jump between offset curves in plane, an adjustment must be taken to add path length for the move between z-steps. Like the planar case, the move between z-steps is taken to be a

path at a 10 degree angle down from the plane, until the next plane is reached. A graphical representation of what these extra segments look like can be seen in red in Figure 21:

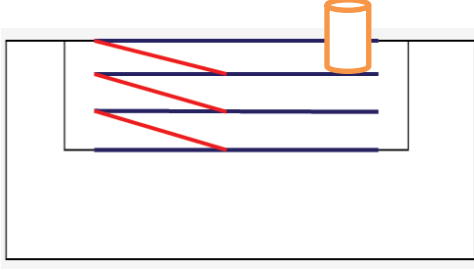


Figure 21: Extra Path Length for Travel Between Z-Steps

For both a big tool and a small tool, if required, we have determined the planar tool path length, the number of z-steps required, and the required tool motion adjustments. At this point we can calculate the total tool path length for both of these tools, which the machining model will use to determine machining time.

**large tool total path length :=**

$$\left( \sum L_{off.big} + \frac{R_{tool}}{\sin(20 \text{ degrees})} * (N_{offsets} - 1) \right) * n_{z.big} + \frac{R_{tool}}{\sin(10 \text{ degrees})} * (n_{z.big} - 1) \quad (16)$$

Where L is the length of all offsets selected for big tool path, R is the radius of the large tool, N is the number of planar offsets selected, and n is the number of z-steps required. The total path length for the small tool is given by equation (13)

**small tool total path length :=**

$$\left( \frac{\text{Area of Face} - \text{Area removed by large tool}}{R_{smalltool}} \right) * n_{z.steps} + \text{finishing distance} \quad (17)$$

Where the area removed by large tool is determined by equation (12), and  $n$ , the number of z-steps, is calculated using equation (15).

We are able to generate a value for the expected total path length of machining without actually generating any form of geometric curve. This has obvious advantages from a computational demand point-of-view. We make several simplifications from real tool path planning, and still achieve the ability to accurately plan a tool path for a large tool to within 5% of the actual estimated tool path. The small tool path total length is slightly less accurate, but still yields very reasonable values for machining time (see Chapter 8 for comparisons).

## Chapter 5: Machining Simulation

For milling operations, the operator must select the speed of the machine (RPM), the feed of the machine (IPM), the step-over (in) and the depth of cut (in). While RPM and feed of the machine are usually chosen by the machinist based on past experience, there are fundamental machining mechanics that drive what these values should be; cutting speed and feed per tooth. For the work presented here, a table containing values for Cutting Speed, Feed Rate, and unit power for a variety of materials is used. It is populated using published data from Walker [14] and El-Hofy [18]. The data in this table is loaded by the software at the beginning of the process, to be accessed by the machining model when needed.

The selection of machining parameters begins with the cutting speed, which is a function of the part material and the tool material. This parameter is given in the literature as a range of possible values, and both a high value and a low value are loaded into the software for each combination of part and tool material. This range of values is reported to produce good cutting results, and the choice of where in this range to place the cutting speed is the machinists' choice. In order to capture the possibilities and implications of operating at any point in this range, AMFA will run the simulation 10 times, using a different value for the cutting speed each iteration. 10 iterations were chosen as a good balance between resolution and computational expense. The cutting speed values for each iteration are assigned using a linear interpolation using 10 points.

$$\textit{Cutting Speed} = CS_{low} + \left(\frac{i}{9}\right) * (CS_{high} - CS_{low}) \quad (18)$$

Where  $i$  is the iteration counter, starting at 0 and counting to 9, incremented by 1 each iteration. Each of these 10 points will result in a different value for machining time, and combined with different values of tool wear for each simulation, will suggest an optimal speed for machining for the lowest cost.

Cutting speed can be translated to a machine RPM if the tool diameter is known using equation 19.

$$RPM := \frac{CS * 4}{D_{tool}} \quad (19)$$

The IPM is the linear advancement of the cutting tool, and is given in units of inches per minute. It is based on feed per tooth, a published value depending on the tool and part materials and tool diameter, and the RPM of the machine. The IPM can be calculated using equation 20.

$$IPM := FR * N_{flutes} * RPM \quad (20)$$

Step-over and Depth of Cut are functions of part and tool material, and tool diameter. For the work presented here 50% is used consistently for both step-over and DOC, although this parameter is very easily adjusted to match machinist preference.

The Material Removal Rate (MRR) is an important machining parameter that determines the power requirement of the machine, and can be found using equation 21.

$$MRR := DOC (in) * Steptover (in) * IPM \left( \frac{in}{min} \right) \quad (21)$$

The power required to achieve a specific MRR is a function of the material being cut, characterized by the materials' Unit Power, a published value. The power requirement can be calculated using equation 22.



$$\mathbf{Spindle\ Power} := \mathbf{MRR}(\frac{\mathbf{in}^3}{\mathbf{min}}) * \mathbf{UnitPower} \left( \frac{\mathbf{HP*min}}{\mathbf{in}^3} \right) \quad (22)$$

Given the IPM setting of the machine, and the tool path length for the given tool, we can determine the machining time for that tool. Naturally, the entire model must execute twice if two tools have been selected for the partition. The total machining time can then be expressed by equation 23.

$$\mathbf{Time}_{\mathbf{machining}} := \frac{\mathbf{Large\ tool\ Path\ length}}{\mathbf{IPM}_{\mathbf{large}}} + \frac{\mathbf{Small\ Tool\ Path\ Length}}{\mathbf{IPM}_{\mathbf{small}}} \quad (23)$$

The extra time associated with using two tools, (i.e. tool change time) will be accounted for when the non-productive production times are determined in Section 5 of this work. The result of the machining simulation can be seen below in figure 22, showing the range of times produced from working with a range of given cutting speeds.

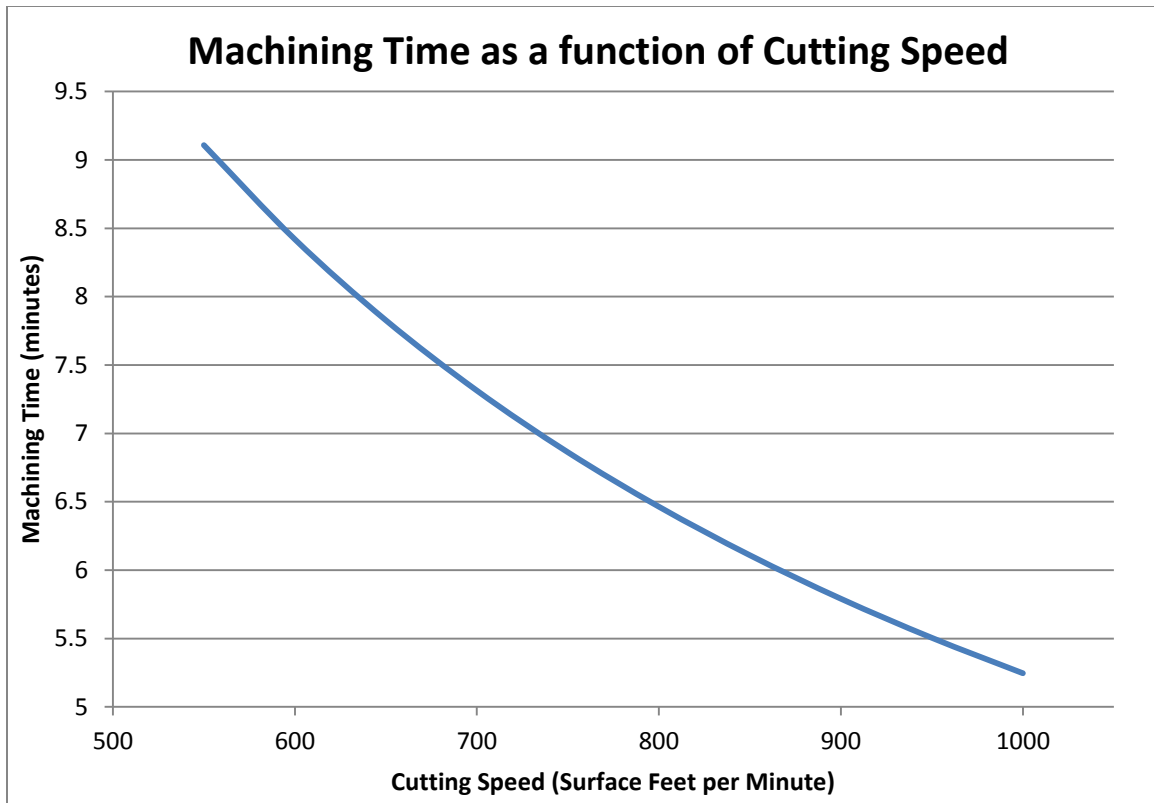


Figure 22: Machining Times vs. Cutting Speed

Since the cost of running the machines is based on known time-driven values, we can easily translate these times to costs associated with running the machines using the parameters of operator rate and labor rate associated with the current machine. The resulting cost as a function of cutting speed can be seen in figure 23:

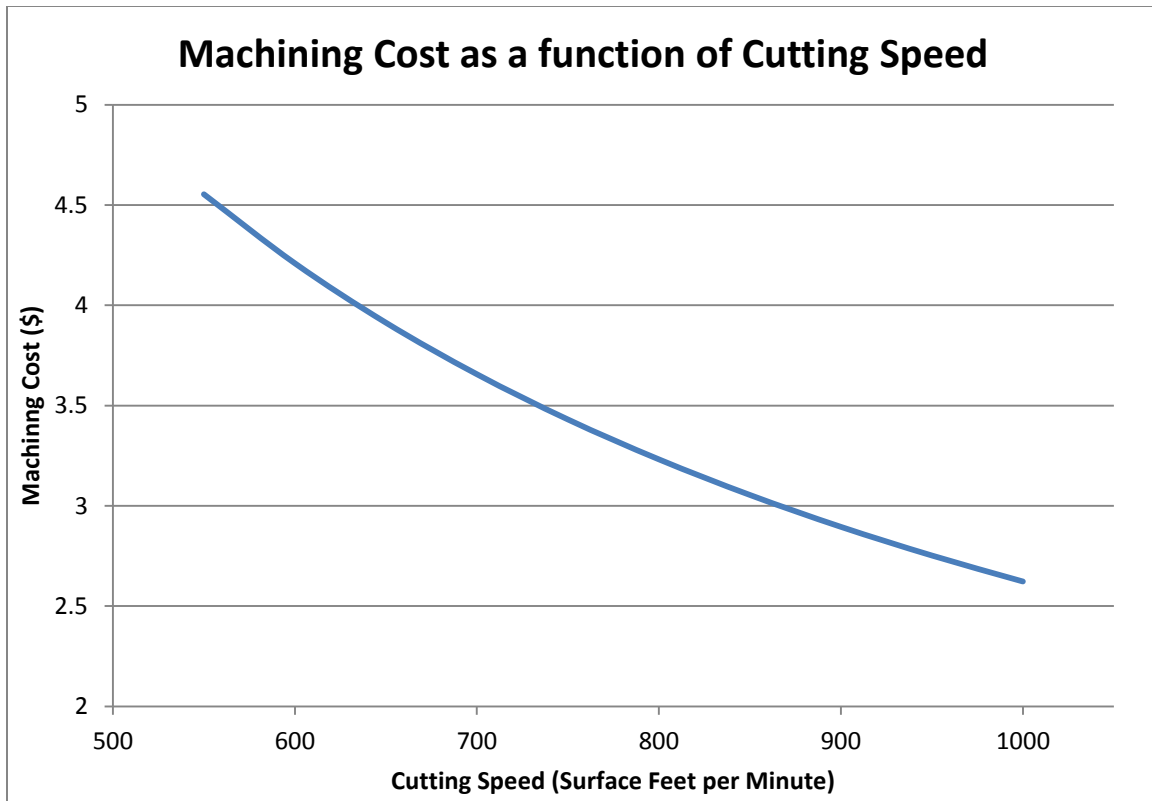


Figure 23: Cost Driven by Machine Occupation Time vs. Cutting Speed

### TOOL WEAR MODELING

In addition to selecting appropriate machining parameters and estimating the time to machine a part based on tool path, there is a clear need to estimate the wear and expended tooling in a given machining operation. If time is the only metric recorded during machining, it will always look profitable to machine at the fastest speed available, since it will take the least time, and machines and foundries typically bill by the hour. Estimating the wear to the tooling involved allows for a second metric to be tracked, where time and cost have an inverse relationship. Literature suggests that there is a specific machining speed that leads to the lowest overall cost for machining, based on the

time of machining and the expense of consumed tooling. The relationship between machining time, tool wear, and overall operation cost can be seen below in figure 24, reproduced from Kalpakjian[19].

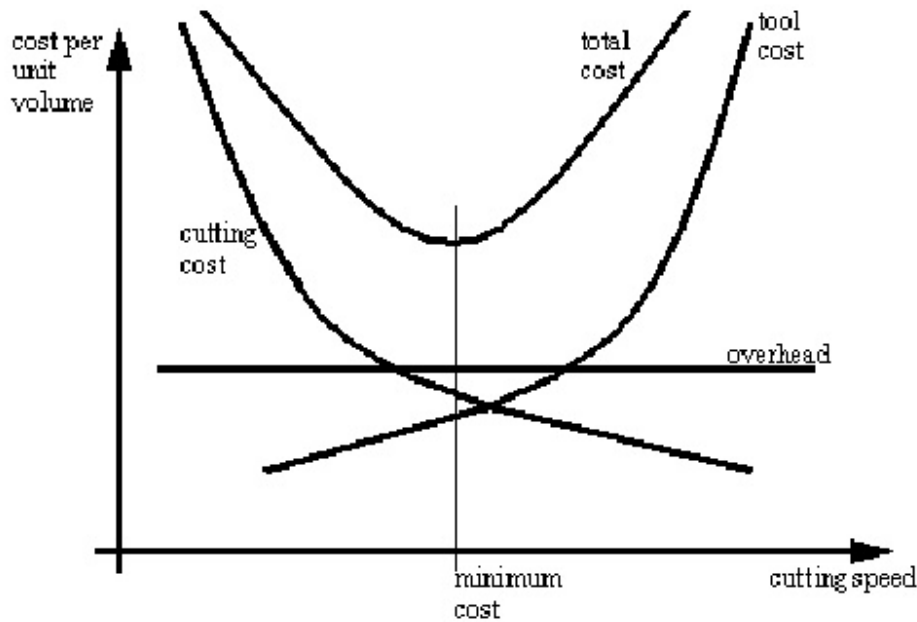


Figure 24: Total Cost for Machining Processes Based on Cutting Speed

The cutting cost line in figure 24 is familiar, and similar to the results produced from running a simulation on AMFA where 10 points are created based on iterating the cutting speed. To generate a curve for tool wear, we need to parametrically model the wear of the tooling used in the machining simulation.

Tool wear modeling was first accomplished by Taylor [20], who generated a famous equation for determining the expected tool life given a range of cutting parameters and experimental constants. Although this equation was developed for turning machining on a lathe. Recent work has validated adapting the model for use in

milling processes [20][21][22]. The form of the equation used here is known as the “expanded” Taylor’s equation, as it includes parameters that have a smaller contribution to the estimation. The form of the equation used is borrowed from [20], and is shown in equation 24:

$$\textbf{Tool Life} = \textbf{C} * \textbf{CS}^{-\textbf{x}} * \textbf{IPM}^{-\textbf{y}} * \textbf{DOC}^{-\textbf{z}} \quad (24)$$

Where Tool Life is given in minutes, CS is the cutting speed, IPM is the feed rate of the machine, and DOC is the depth of cut of each pass. C,X,Y, and Z are all experimentally determined constants. These constants depend on many parameters, including the fixture, the machine, overall part stiffness, etc., but are primarily functions of the tool itself and the part material. These constants are typically determined by a shop in a particular setup that will be running the identical setup for a large volume of machining. Since we lack the ability to generate real values for these constants, we request these parameters be loaded with the tool information, since the tool has the largest effect on the tool life. For current simulations we are using generally suggested “typical values” of these parameters, as suggested by Kalpakjian [19], and Kibbe[23]. For long term improvement of accuracy, a shop using AMFA would be expected to measure and record these values for their particular setup, so that AMFA could provide more accurate feedback about the optimal machining speeds. The proof-of-concept parameters we are using are currently set at:

$$\textbf{C} = 3.35 \text{ E}10, \textbf{X} = 2.93, \textbf{Y} = 1.15, \textbf{Z} = .57$$

Given that X,Y,Z are exponential coefficients in the tool life equation, from the values of these constants it is clear that tool life is most effected by cutting speed, less effected by feed rate, and least effected by the depth-of-cut. Other parameters have been found to have a very small effect on tool life, but are typically negligible, and are ignored here and in most models of tool wear.

For each iteration through the machining simulation with a given cutting speed, the estimated tool life in minutes will be determined using equation 24. The cost of tooling can then be found using equation 25 below:

$$\textbf{Consumed Tooling Cost} = \frac{\textit{machining time}}{\textit{tool life}} * \textit{tool purchase cost} \quad (25)$$

Dividing the machining time by the tool life gives the amount of tooling consumed as a fraction of a tool. I.e. if the machining tool 10 minutes, but the estimated tool life is 100 minutes, we consumed 0.1 tools. If the tool cost \$20, there was \$2 worth of tooling consumed in this machining process.

This tooling cost is calculated for each of the 10 machining times pertaining to the changing cutting speeds. An example of the resulting range of values can be seen in Figure 25, the result of running the highlighted partition seen in figure 7.

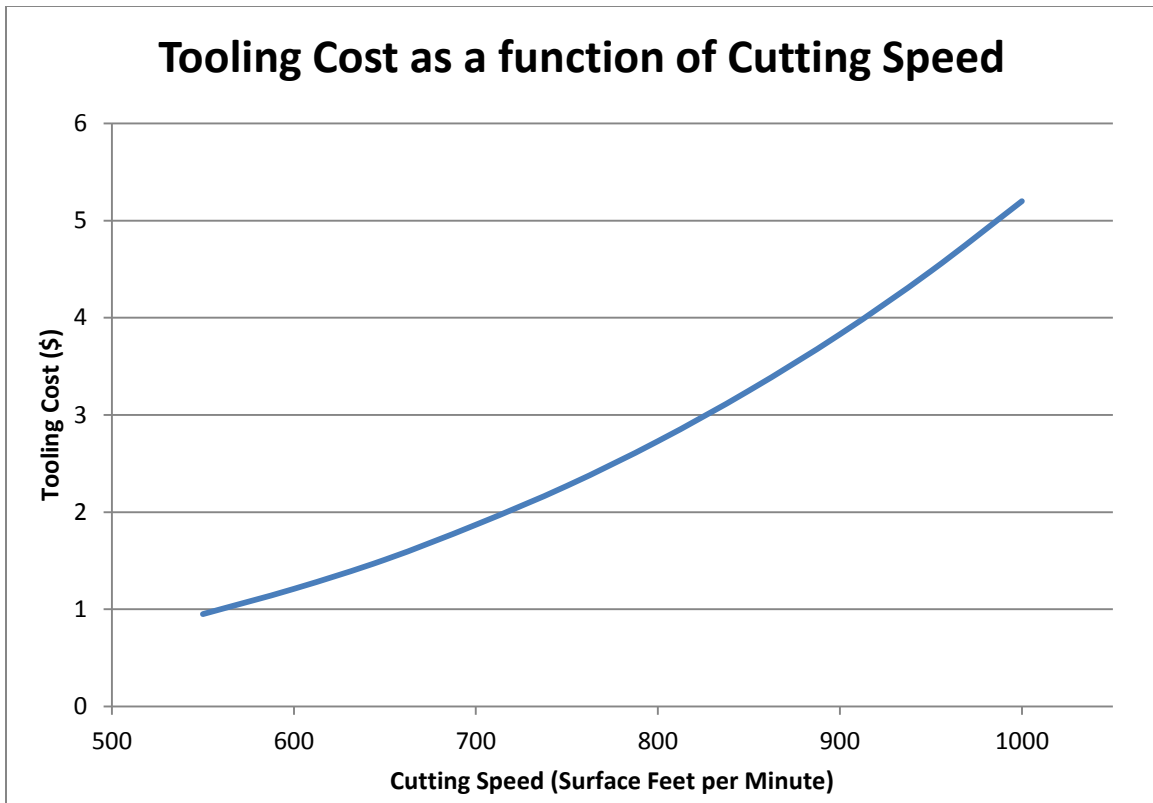


Figure 25: Tooling Cost vs. Cutting Speed

It is clear that machining time and tool wear have inverse relationships with cutting speed. This is the desired effect, and will give us the clear optimal cutting speed for performing the operation at the lowest cost. The costs of machining and tool wear are combined, and the results are shown in figure 26.

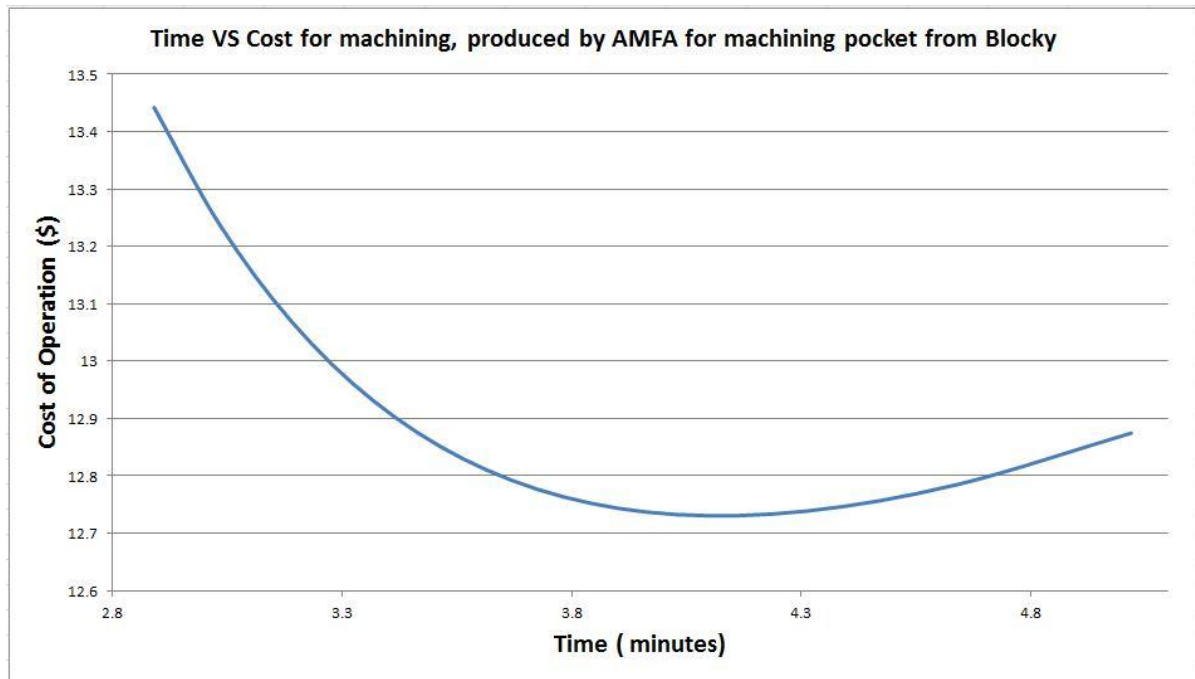


Figure 26: Total Cost of Performing Operation with Varying Cutting Speed

From figure 26, there is a clear point where the cost of the operation is at a minimum. This curve represents what is shown in the literature, (figure 24 above). Despite finding this minimum cost, we want to maintain the rest of the curve as possibilities for machining, as the machinist may still have a need to cut the time of the operation down, even if at a higher cost, i.e., running late on delivery, etc. At a later time we will be interested in presenting the operator of the software with a lowest cost option and a lowest time option, which gives a good indication of the range available.



## **Chapter 6: Non-Machining Production Time**

For any machined part, particularly if it requires multiple processes to complete, the actual machining time does not accurately depict the time required to create the part. Non-productive times can account for as much as 66% of the time [4]. Luckily, since AMFA follows a part through the entire process plan, we have the opportunity to capture and quantify these non-productive part processing times, which occur between operations.

AMFA breaks down a process plan in the following way. The evaluation module estimates time and cost for a machining operation. Each partition is assigned to a single operation. A process plan is then the set of operations that remove all partitions, resulting in the completed part. By looking at the plan, we are able to analyze the difference between adjacent operations and assign times accordingly.

Along with the machining time estimation, for each operation, the evaluation method will also compare pertinent information for the current operation, and the previous operation. There are a number of parameters that may change between operations, which can each be represented as non-machining production times. These times are broken into three categories: fixture time, tool change time, and machine switch time.

Fixture time is the amount of time required to fix a part in a machine so that it may be machined. This value can vary greatly, based on part size and complexity, as well as the method of fixturing. At this time we are unable to predict these parameters based on part and facility data, and are using a fixed time value for fixturing. Future efforts will focus on expanding this functionality. There are two checks that need to be performed to determine the fixture time: “first operation”, or “orientation changed”.

The first operation simply checks to see if this is the first operation that is being performed in the process plan for this part. If this is true, the fixture time is assigned, since the part will always need to be fixtured. If it is not the first operation, we simply check to see if the machine or orientation has changed since the previous operation. If the part has been moved to a different machine, or it is on the same machine but has been re-oriented, it is assigned another time value, composed of summing an unfixturing time, inspection and deburring time, and a fixture time. A flowchart of this process can be seen in Figure 27:

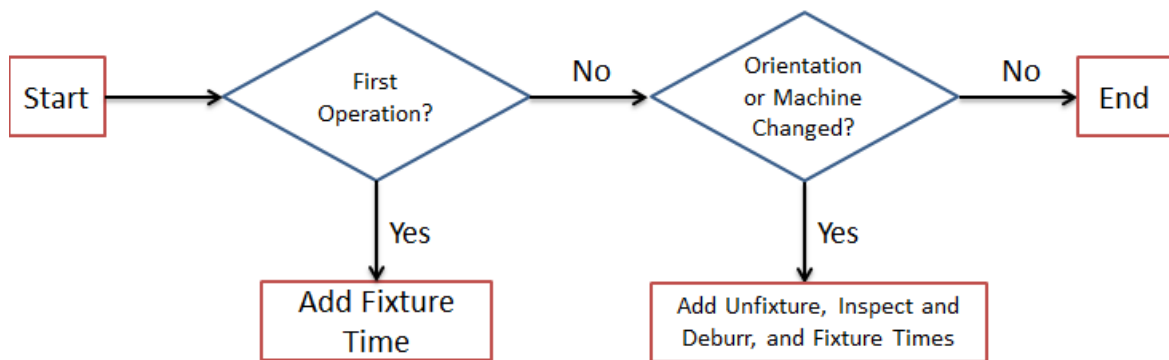


Figure 27: Flowchart of Fixture Time Assignment

Tool change times will contribute only a small amount to the overall process plan time and cost, since many modern CNC machines have tool change times of only a few seconds. However, for completeness, each tool change and the corresponding time delay in processing is recorded into the process plan. The tool change time can become more important when dealing with a machine that does not automatically change tools, as the time for an operator to perform the switch could be greater than a minute. Recall that the tool selection method is able to select two tools for a single operation. If this is the case,

we will add the tool switch time of the machine to the operation here, in the tool switch time method. The tool switch time of the machine is also added when the machine or tool is different from the previous operation, to account for the extra time in switching operations due to setup of the tool in the machine. A flowchart for assigning the tool change time (TCT) can be seen in figure 28.

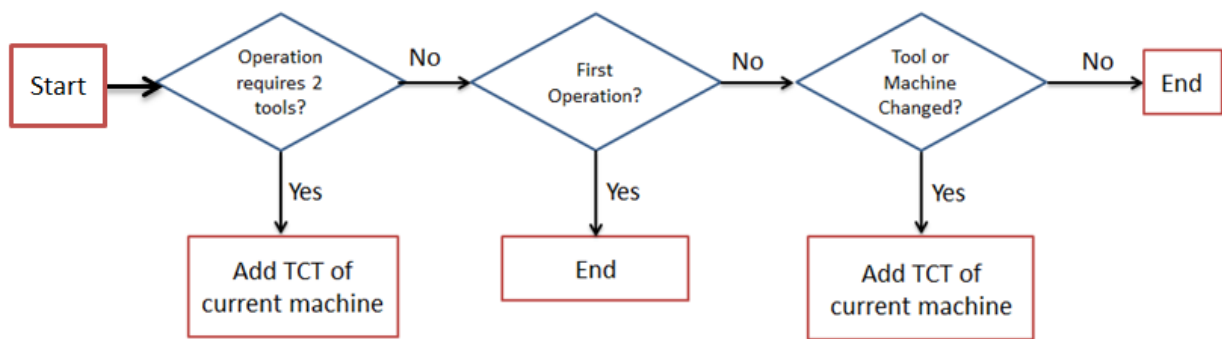


Figure 28: Flowchart of Tool Change Time assignment

The machine switch time module is intended to capture the effect of the shop floor layout, where it takes time to move the part between machines required for consecutive operations. Currently this is a single value assigned when the current machine is different than the previous machine, since we are not working with any particular layout of machines. If the layout were known, this time would be a function of the walking distance between machines, and the size/weight of the part. Currently we are researching this method to include the parameters previously listed. A flowchart of the machine switch time assignment can be seen in figure 29.

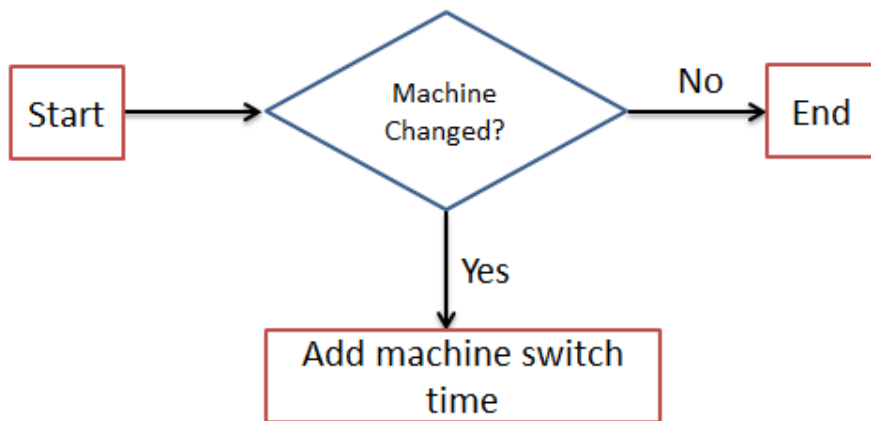


Figure 29: Flowchart of Machine Switch Time Assignment

## Chapter 7: Determining Operation Cost

At this point we have accounted for all of the time periods required to make the part. To translate this list of times to cost, the rates for the shop that is making the part are required. Currently these are defined as machine rate and operator rate. Machine rate is intended to be the cost charged to operate a machine, free of operator. The operator rate captures the charge for the labor performed by the shop floor staff. This includes the time they spend on machine setup, inspection, etc. For many shops these may be a single value, as an hourly charge for however long it takes to make the part. We have chosen to keep them separate, to allow for shops that choose to charge different rates depending on what machines are required and how much operator time is involved. We can now take these two cost rates, the machining time, fixture time, tool change time, machine switch time, and calculate the total cost to perform the given operation, using equation 26.

$$\text{Operation Cost} := \text{Rate}_{\text{machine}} * (t_{\text{machining}} + t_{\text{toolswitch}}) + \text{Rate}_{\text{operator}} * (t_{\text{fixture}} + t_{\text{machineswitch}}) \quad (26)$$

Equation 26 will estimate the total cost of an operation, including non-machining times taking place between the current operation and the previous operation. Once an entire process plan has been generated, the cost of each operation in the plan can simply be summed to estimate the total production cost of the part.

## Chapter 8: Simulation Results

Preliminary results for 3 distinct machining situations are presented here briefly: internal pocket single tool, internal pocket double tool, and external pocket double tool. A discussion of a fourth situation involving adjacent partitions is also presented.

### 8.1. INTERNAL POCKET SINGLE TOOL

For an internal pocket that is concave in nature, the negative of the shape is convex, therefore the negative will not be reduced by convex decomposition, and the entire pocket will be evaluated in a single operation. A solid model composed of a cube with an interior pocket in the top was generated, and used for both AMFA analysis, and loaded into FeatureCAM, a commercial CAM package, used as a reference of accuracy. The part used can be seen below in Figure 30:

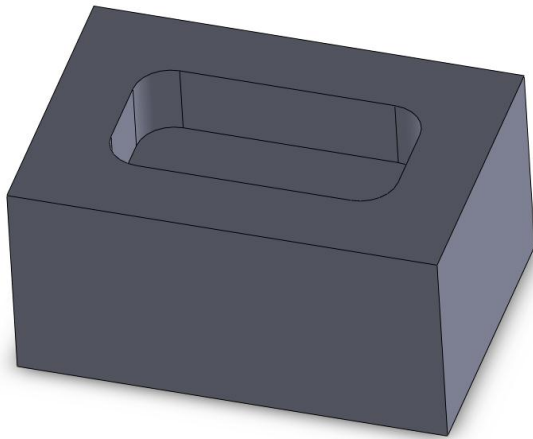
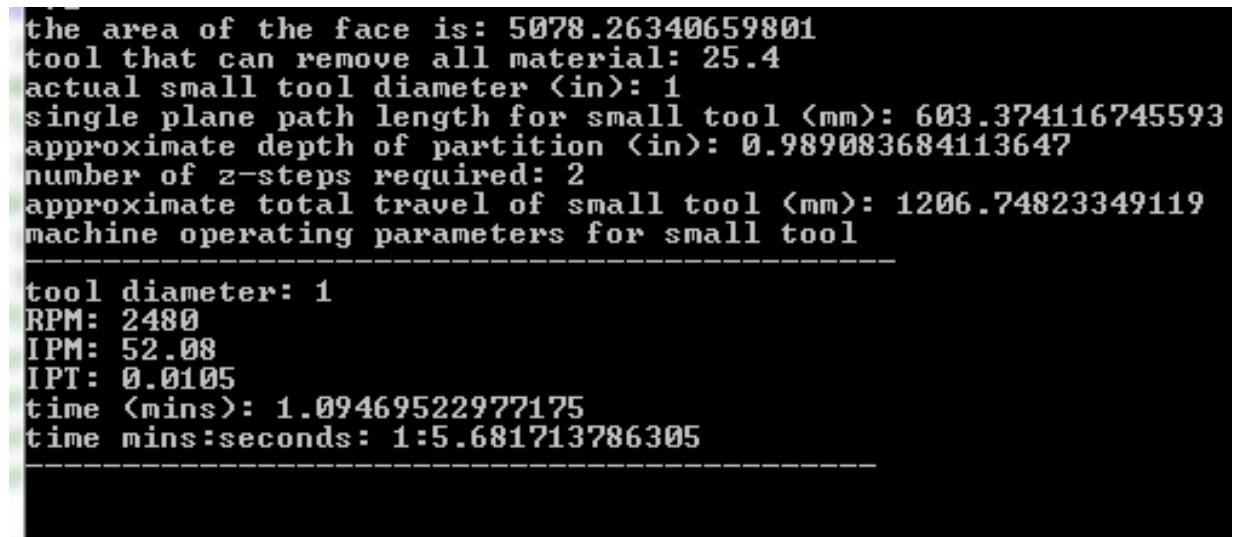


Figure 30: Single Tool Internal Pocket Part Used for FeatureCAM and AMFA Comparison

When FeatureCAM is asked to remove the pocket with a single tool, it generates the following parameters:

**Op: 1 side1 (rough1)**  
**F/S: 2482 RPM, 49.7 IPM (0.0100 IPT)**  
**Tool: #1 (endmill1000:reg, 1.0000 in.)**  
**Depth: 1.0000 in. (in 2 steps, 0.5000 in. each)**  
**Other: Stepover: 0.5000 in.**  
**Time: 1:07.1**  
**Power: 3.72 (est. 3.72) HP**

AMFA return the following window:



```
the area of the face is: 5078.26340659801
tool that can remove all material: 25.4
actual small tool diameter <in>: 1
single plane path length for small tool <mm>: 603.374116745593
approximate depth of partition <in>: 0.989083684113647
number of z-steps required: 2
approximate total travel of small tool <mm>: 1206.74823349119
machine operating parameters for small tool
-----
tool diameter: 1
RPM: 2480
IPM: 52.08
IPT: 0.0105
time <mins>: 1.09469522977175
time mins:seconds: 1:5.681713786305
-----
```

Figure 31: AMFA Return for Single Tool Pocket

This part is the simplest case for tool path planning and time estimation, still the results are promising. FeatureCAM predicts a machining time of 1:07.1, while AMFA predicts 1:05.7. For the machining model, the selected parameters are nearly identical, the largest disparity coming from the IPM (linear feed) of the machine (FeatureCAM –

49.7, AMFA – 52.08 ). For a slightly more difficult case, the same pocket was changed to have much smaller corner radii, necessitating the use of multiple tools to keep the machining time down.

## **8.2. INTERNAL POCKET DOUBLE TOOL**

Using this type of part with AMFA allows for the validation of the accuracy of the tool selection method, the machining model, and the tool path planning method. The part used for both FeatureCAM simulations and AMFA is shown in Figure 32.

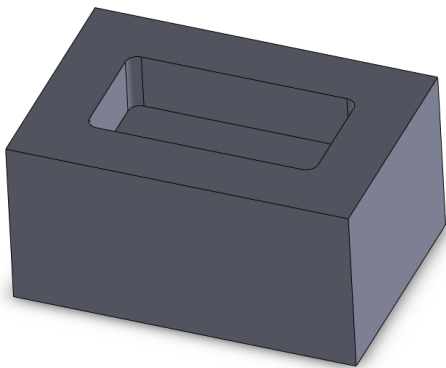


Figure 32: Two Tool Internal Pocket Part Used for FeatureCAM and AMFA Comparison.

Note the small interior corners on the part. To simulate this part in FeatureCAM, no additional information was provided beyond the machine start face and the feature to be removed. By doing this, the software was required to determine if more than one tool



was ideal, and select sizes for both tools. When the simulation was performed, FeatureCAM returned the following machining parameters.

**Op: 1 side1 (rough1), Fixture 1**  
**F/S: 2482 RPM, 49.7 IPM (0.0100 IPT)**  
**Tool: #1 (endmill1000:reg, 1.0000 in.)**  
**Depth: 1.0000 in. (in 2 steps, 0.5000 in. each)**  
**Other: Step-over: 0.5000 in.**  
**Time: 1:07.3**  
**Power: 3.72 (est. 3.72) HP**  
**Op: 2 side1 (rough2), Fixture 1**  
**F/S: 5200 RPM, 32.5 IPM (0.0031 IPT)**  
**Tool: #2 (endmill0312:high+, 0.3125 in.)**  
**Depth: 1.0000 in. (in 7 steps, 0.1429 in. each)**  
**Other: Step-over: 0.1563 in.**  
**Time: 2:04.9**  
**Power: 0.22 (est. 0.22) HP**

Note that Op:1 corresponds to a large roughing tool, and Op:2 is the small tool that will machine the corners and perform one finish pass around the perimeter of the pocket. AMFA returns the screen (shown in Figure 33).

```

the area of the face is: 5197.84384198969
two tools will be required for this partition
small tool diameter (mm): 10.16
large tool diameter (mm): 25.5
actual small tool diameter (in): 0.375
single plane path length for small tool (mm): 472.984396774471
actual big tool diameter (in): 1
single plane path length for big tool (mm): 319.188
approximate depth of partition (in): 0.988703730408411
number of z-steps required: 6
=====
approximate total travel of small tool (mm): 2837.90638064682
machine operating parameters for small tool
=====
tool diameter: 0.375
RPM: 6613.333333333333
IPM: 59.52
IPT: 0.0045
time (mins): 2.25259269482381
time mins:seconds: 2:15.1555616894284
=====
approximate total travel of big tool (mm): 1230.52059635033
machine operating parameters for large tool
=====
tool diameter: 1
RPM: 2480
IPM: 53.32
IPT: 0.01075
time (mins): 1.09030066248383
time mins:seconds: 1:5.41803974902958
=====

```

Figure 33: AMFA Return for Two Tool Pocket

Note that AMFA is returning the small tool operation first, then the roughing tool. The way the evaluation method simulates each tool, it does not matter which operation is simulated first. Starting with the roughing cut, both packages: selected the same large tool (1" diameter), estimated machining time as 1:06 +/- 1 second, and had all machining parameters similar. For the smaller tool, the size of the tool selected varies slightly. FeatureCAM selects a .3125" endmill, while AMFA selects a .375" endmill. Since the radius of the corners is 0.4", both of these tools will work well. AMFA generates machining parameters for this tool much higher than FeatureCAM: 6613 RPM vs. 5200 RPM, 59 IPM vs. 32.5 IPM. The time for machining is similar; FeatureCAM predicts 2:05, while AMFA predicts 2:15. Since AMFA selected more aggressive machining

parameters, but still estimated a slightly longer machining time, it has planned a longer tool path than FeatureCAM for this simulation.

### **8.3. EXTERIOR POCKET DOUBLE TOOL**

Next, we compare the accuracy of the tool for an open-boundary machining case. The part used for this simulation is shown in figure 34.

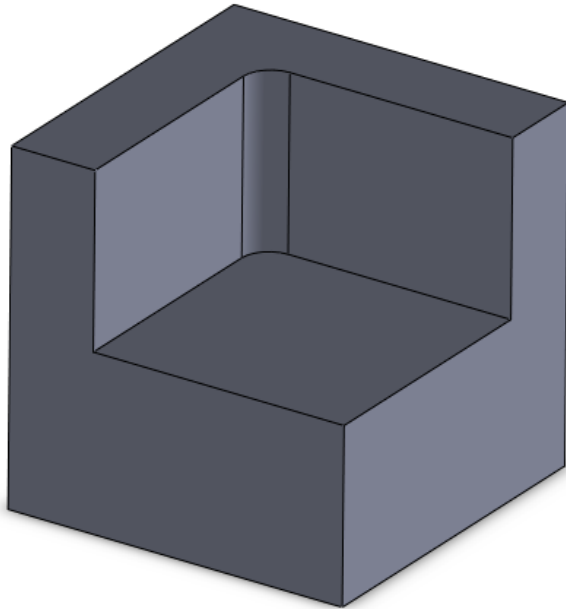


Figure 34: Double Tool External Pocket Used for FeatureCAM and AMFA Comparison

Once again, a small internal fillet should force both software packages to select two tools for machining. Like the previous cases, this part was provided to FeatureCAM

with only the machine start face determined. The following results are achieved with FeatureCAM:

**Op: 1 side1 (rough1), Fixture 1**  
**F/S: 2482 RPM, 49.7 IPM (0.0100 IPT)**  
**Tool: #1 (endmill1000:reg, 1.0000 in.)**  
**Depth: 1.5000 in. (in 3 steps, 0.5000 in. each)**  
**Other: Step-over: 0.5000 in.**  
**Time: 1:46.5**  
**Power: 3.72 (est. 3.72) HP**

**Op: 2 side1 (rough2), Fixture 1**  
**F/S: 4965 RPM, 49.7 IPM (0.0050 IPT)**  
**Tool: #2 (endmill0500:reg, 0.5000 in.)**  
**Depth: 1.5000 in. (in 6 steps, 0.2500 in. each)**  
**Other: Step-over: 0.2500 in.**  
**Time: 0:30.0**  
**Power: 0.93 (est. 0.93) HP**

Note that Op:1 is a roughing tool to remove the pocket, and Op:2 is cleaning up the small radius and finishing the two walls. AMFA returns the screen shown in Figure 35.

```

actual big tool diameter (in): 1
single plane path length for big tool (mm): 390.003606081449
approximate depth of partition (in): 1.47449065757559
number of z-steps required: 6
=====
approximate total travel of small tool (mm): 1014.40738373589
machine operating parameters for small tool
=====
tool diameter: 0.5
RPM: 4960
IPM: 57.04
IPT: 0.00575
time (mins): 0.840195622137708
time mins:seconds: 0:50.4117373282625
=====
approximate total travel of big tool (mm): 1847.61177588081
machine operating parameters for large tool
=====
tool diameter: 1
RPM: 2480
IPM: 53.32
IPT: 0.01075
time (mins): 1.63707324300832
time mins:seconds: 1:38.2243945804992
=====

```

Figure 35: AMFA Return for Two Tool Open Pocket

Comparison of the FeatureCAM and AMFA results show that both packages selected the same two tool sizes. For the large tool, chosen parameters are very similar, and the time varies by only a few seconds (1:46.5 for FeatureCAM vs. 1:38.2 for AMFA). The smaller tools, however, have a larger difference in required machining times. FeatureCAM requires 30 seconds for the small tool, while AMFA predicts 50 seconds. While this is not an enormous difference overall, it is significantly off the accuracy of the large tool simulations. While the finishing distance adjustment allows us to quantify the distance travelled to “clean up” for this tool, we still overestimate the amount of material that this tool has to remove.

The area remaining for the small tool to machine will be incorrect. The large tool will not remove all material from any of the corners. However, only one of these corners (inside pocket) is a “real” corner. The area calculations done in AMFA will leave all of

these corner residuals for all 4 corners for the small tool, overestimating the time for the small tool.

Overall, AMFA is very accurate for interior pocket simulations, when all boundaries of the face are real boundaries. This assures is that the tool selection, tool path length, and machining models are reasonably accurate. The next stage of research in this area is to add the ability to reason about the number of virtual corners in the starting face, so that more accurate values for the area required for machining by the small tool could be more accurate.

#### **8.4 ADJACENT PARTITIONS**

One situation of interest is when we have multiple adjacent partitions to machine, which a feature recognition engine would treat as a single partition. An example of such a part is shown in Figure 36.

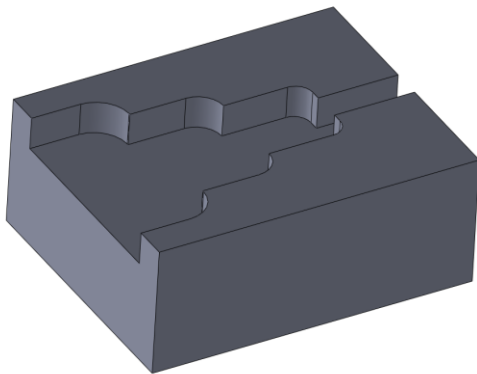


Figure 36: Adjacent Partitions Part

FeatureCAM recognizes the entire pocket as a single feature, and selects tools and plans tool paths that will machine the entire area in a single operation. AMFA, on the

other hand, due to convex decomposition, will see the pocket as 4 distinct partitions, and therefore, 4 operations. Each of these partitions will have its own tool selection process and path planning. To AMFA, the part may look like the diagram in Figure 37(a).

Since all partitions shown in Figure 37(a) can be removed by a part in a single orientation and fixture, the best process plan will be setting the part up in a machine once, and removing all partitions without moving the part. The optimization in AMFA will return this as the best result, since it will have the best time (no re-fixture times, etc.). However, it can be seen that the tool paths are dramatically different in AMFA for this part compared to FeatureCAM. The only part of the tool path that is important is the length, since it is the driver of machining time. If we examine one of the partitions from the model in more detail, we can compare the length of the tool paths contained within that partition from both packages. The selected partition is shown in Figure 38.

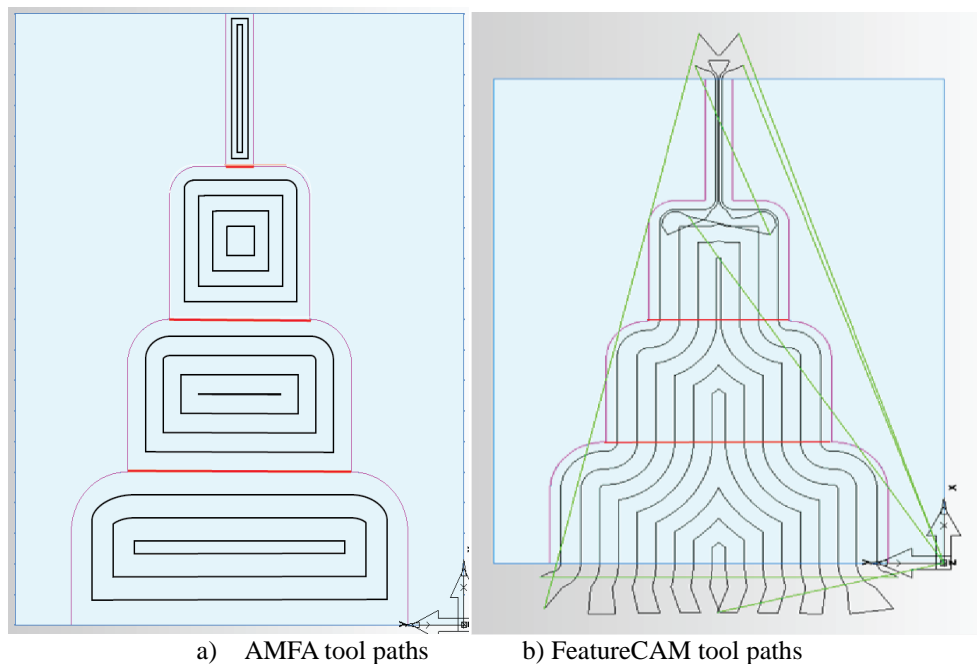


Figure 37: Tool Path Comparison for Adjacent Partitions Example

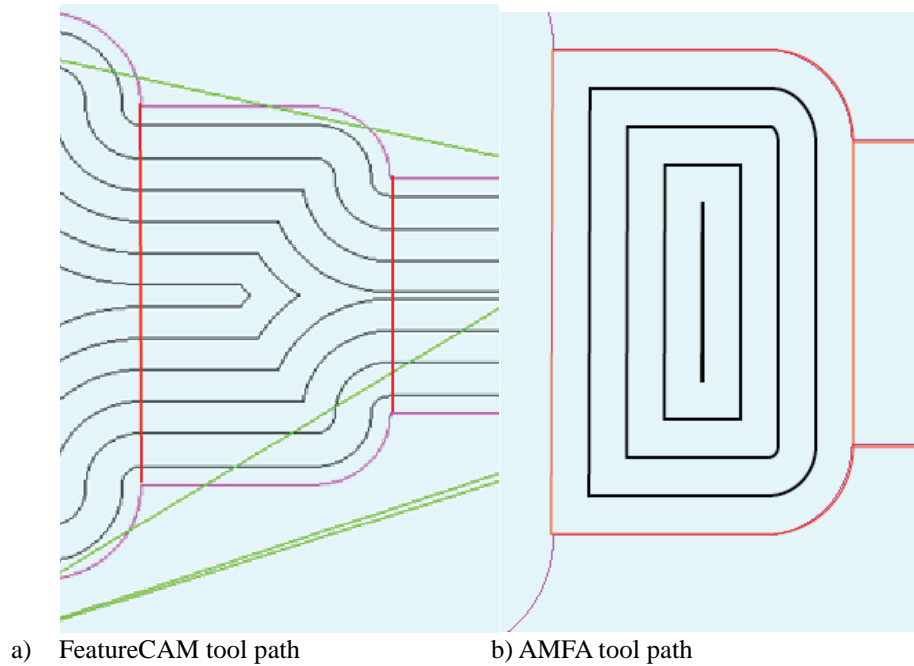


Figure 38: Single Partition Tool Path Comparison for Adjacent Partitions Example

Note in Figure 38(a) we are cutting the toolpaths at the point where convex decomposition sets the boundaries of the partition. Calculating the length of curves within the partition for each of the tool paths in 38(a) and 38(b), results in the following:

**FeatureCAM : 748 mm**  
**AMFA : 739 mm**

This represents a difference of around 1% in the tool path length contained within the partition. AMFA may still overestimate the time to machine the entire part due to inaccurate non-machining tool travel. Important to note is that these paths are generated assuming the same tool size. Another complication AMFA faces is the fact that tools are selected for each of the four partitions, where FeatureCAM uses only 2 for the entire feature. The ability to reason about tooling sizes used in adjacent partitions would further improve the accuracy on parts like this.



## **8.5. CONCLUSIONS AND FUTURE WORK**

Although the results shown previously demonstrate AMFA's ability to simulate the machining of partitions very well, there is an obvious element missing: the computation time comparison. Although AMFA feels much faster in generating tool paths than FeatureCAM, we are forced to operate featureCAM through the native graphical user interface, and have not been able to accurately clock the computational speed of the software. We are currently investigating CAM packages that can be operated remotely through an API, which will allow us to accurately clock the time requirements of these packages against our own capabilities, which will further validate the efforts presented here.

All of the work presented previously allows AMFA to evaluate individual partitions. A search process can create a machining plan that provides an option for removing every partition in the negative, resulting in the completed part. Additional processing of this machining plan is required, and will be presented in Chapters 10-13.

Currently AMFA is being expanded to include a variety of manufacturing processes, including turning, drilling, welding, sheet cutting, and sheet bending. Functionality for all of these processes will be built into the presented evaluation module.

The ability to reason about the shape and geometry of the partition has proven useful in calculating the small tool path length, and it is likely there are several other ways we could use this information to improve the accuracy and abilities of the model.

A more advanced model of the non-machining production time is being produced, to more carefully evaluate fixture time for parts, part movement between machines, and other times (inspection, deburring, etc.).

## SECTION 2: MACHINING PLAN PROCESSING

The search process of AMFA is able to generate a machining plan to remove all necessary partitions in order to create a part. As mentioned previously, the structure of AMFA is that a single machine operation is related to the removal of a partition. A machining plan is then a list of machine operations. The structure of a machining plan is shown in figure 39.

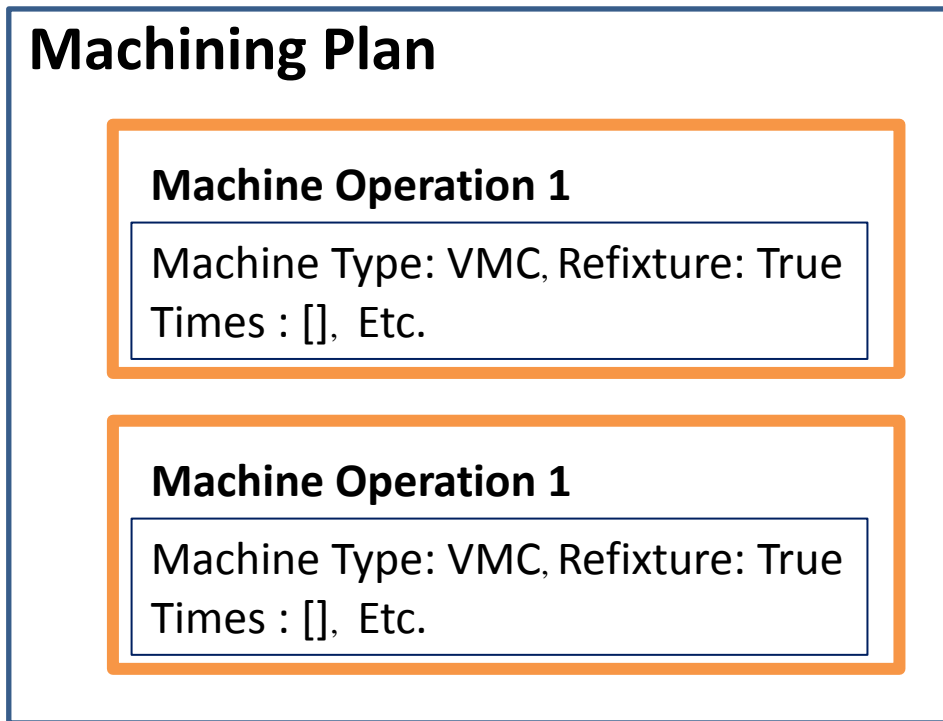


Figure 39: Structure of a Machining Plan

Each machine operation has been analyzed following the procedure outlined in Chapters 3-7. However, this analysis has been performed without considering actual machines as part of the machining simulation model. Using actual machines will affect the time and cost of an operation, since the capabilities of the machine may dictate a deviation from the calculated machining parameters.

In addition to considering the use of actual machines for each operation, the machining plan can sometimes be simplified, so that the plan is easier to read and understand, and there are fewer options for machining. The overall goal is to return to the user an easy to read and understand machining plan which dictates the use of specific machinery in the plan. In order to accomplish this task, several additional modules have been developed. These modules are machine selection, machining options generation, machining plan simplification, candidate generation, and candidate filtering.

## **Chapter 9: Machine Selection**

Automated Manufacturing Feedback Analysis is meant to operate with information about the operator's foundry. Therefore, AMFA has access to a database which contains all of the machines available in the foundry, along with the pertinent parameters of each of these machines that will be required for the machining simulation to be performed.

For each type of machine considered by the representation schema of AMFA, a list of available machines of that type in the database are created at the beginning of the software execution. In order to be added to the list of available machines, the machine must pass a prequalification check that ensures the part can be made in that machine. This check is based solely on machine part size limitations (table size and spindle clearance) and the bounding box dimensions of the given part. In rare instances, the table size of a machine far exceeds the axis travel capabilities, meaning that a part that fits on the table may not necessarily be able to be machined completely, as the limited travel of the machine would prevent the spindle from accessing the extreme boundaries of the part. The size of the area requiring machining on a part is named "feature size", and provides an additional parameter with which to disqualify machines. However, the computation of feature size for a part is non-trivial, and relevant only for the simplest of knee-mill machines with limited table travel. A summary of the part parameters considered and their relevance for machine selection can be seen in table 3.

	<b>part size</b>	<b>feature size</b>	<b>part weight</b>
<b>acceptable as only comparison?</b>	yes, for some machines	never	never
<b>complexity to compute</b>	low	moderate	low
<b>likely to be useful?</b>	yes	only for some machines	rarely

Table 3: Part Parameters Relevance for Machine Selection

For the vast majority of machines used in industry, the only parameter of a part that disqualifies a specific machine is the overall part size. Many mills, especially CNC mills, are fully contained systems, and the internal dimensions of the cavity are given. These are the greatest dimensions that a part may have and still be acceptable for machining.

To perform the part size check, a few parameters from both the machine and the part must be found and compared. These parameters are summarized in table 4.

	<b>part parameters needed</b>	<b>machine parameters needed</b>
<b>Bounding box of part</b>	dimension (X)	table (X)
	dimension (Y)	table (Y)
	dimension (Z)	table to spindle (Z)

Table 4: Required Parameters from Part and Machine for Machine Compatibility Check

Table 4 shows that the bounding box of the part is loaded with dimensions assigned to the X,Y,and Z axes. The dimensions can be mapped to axis due to the original orientation of the solid model being preserved in the STEP model. However, the orientation given is not necessarily the orientation in which the part will be fixtured or machined. For a complex part, the orientation of the part may change several times. At this point we do not understand the part features well enough to reason about which part orientations will be desired, so we simply check all possible orientations of the part, and if there is an orientation that will allow the part to be fixture in a given machine, that machine can be added to the list. Mills rarely have square tables, so it is likely that there are parts that will fit into the machine in only specific orientations.

To orient the largest dimensions of the part with the largest dimensions of the mill, the X,Y, and Z of both the part and the machine are loaded into lists, and then sorted in descending order. For a given list index, the element of each list can be compared, and if the value in the part list is greater than the machine list, the part will not fit in the machine, and this machine can be skipped. If all elements of the part array are less than the corresponding element of the machine array, the part will fit in the machine, and this machine can be loaded as a machine available to make the part. An illustration of this process can be seen in Figure 40:

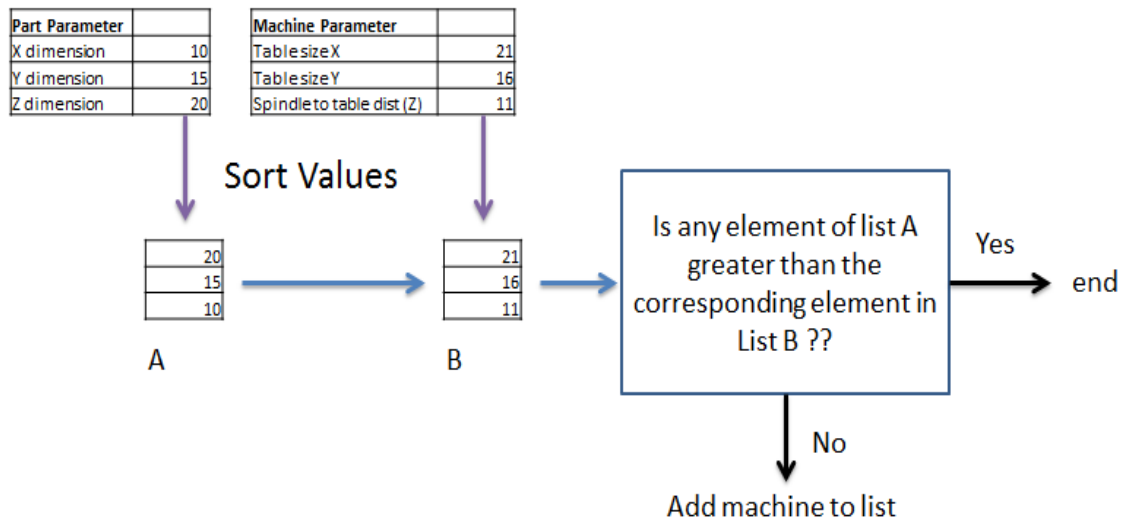


Figure 40: Comparing Part Size with Machine Maximum Part Size

Once a machine is selected as an option for making the part, there are several parameters of the machine that must be loaded for use by the machining simulation module. For each machine, the parameters that must be loaded are:

- Horsepower
- Feed Rate (Inches per minute)
- Labor rate (dollars per hour)
- Maximum RPM (revolutions per minute)
- Tool change time (seconds)
- Operator rate (dollars per hour)
- Machine Name

Currently this process is in operation only for horizontal (HMC) and vertical (VMC) milling centers. AMFA is beginning to process abrasive sheet cutting processes,

as well as drilling, and these machines will need to be added to the database and loaded in a similar fashion. Further development will likely see the addition of turning, grinding, and other typical foundry metal processing equipment into the software. All of these machines will require not only a different validation check, but also a different set of relevant parameters to be loaded.

To analyze operations being performed on specific machines, we need to check the machining parameters selected by the model presented in Chapter 5 against the capabilities of the machines, loaded here. If the machine has the capability to machine with the parameters calculated, all machines will return the same machining time for a given partition, as all machining parameters will be the same. It is a common occurrence, however, for the calculated parameters to exceed the ability of the machine. For small tooling, high RPM's are required, and many tools less than 1/4" will result in exceeding machine capability for RPM. Likewise, very large tools will frequently exceed machine horsepower requirements at the calculated feed rate. In order to capture realistic machining environments for specific machines, a series of checks is implemented to ensure the parameters are within an acceptable range for the given machine.

We can directly compare RPM, IPM, and HP against the abilities of the machine. Comparisons are implemented in the method to check a parameter immediately after it is calculated, to limit error propagation. The checks for the three relevant parameters are given, in order of application, below.

- If the RPM exceeds the machine maximum, the cutting speed is reduced until the RPM value is acceptable



- If the IPM exceeds the machine maximum, the feed per tooth (FR) is reduced until the IPM value is acceptable [14].
- If the spindle power exceeds the machine maximum, reduce feed per tooth (FR) until the HP value is acceptable.

Although literature suggests the reduction of FR to lower power requirements, augmenting the step-over, DOC, CS, or RPM is being investigated as an alternative solution.

## Chapter 10: Machining Options

For each machine operation given in the machining plan, that operation can be performed on any of the machines available of the machine class that is specified in the operation. The lists of available machines of each type have been formed previously, as described in Chapter 9. For each machine, a new instance of the operation is created, and all the models described in section 1 are performed for the operation. Where the differences in the resulting time and cost come from are due to the machine limitations as described in Chapter 9, where some machines may require the machining parameters to be augmented, resulting in a longer machining time. This, along with varying billing rates for each of the machines, will lead to a variety of ways to machine a partition with diverse times and costs. A diagram demonstrating the process of translating an operation to a range of options can be seen below in figure 41:

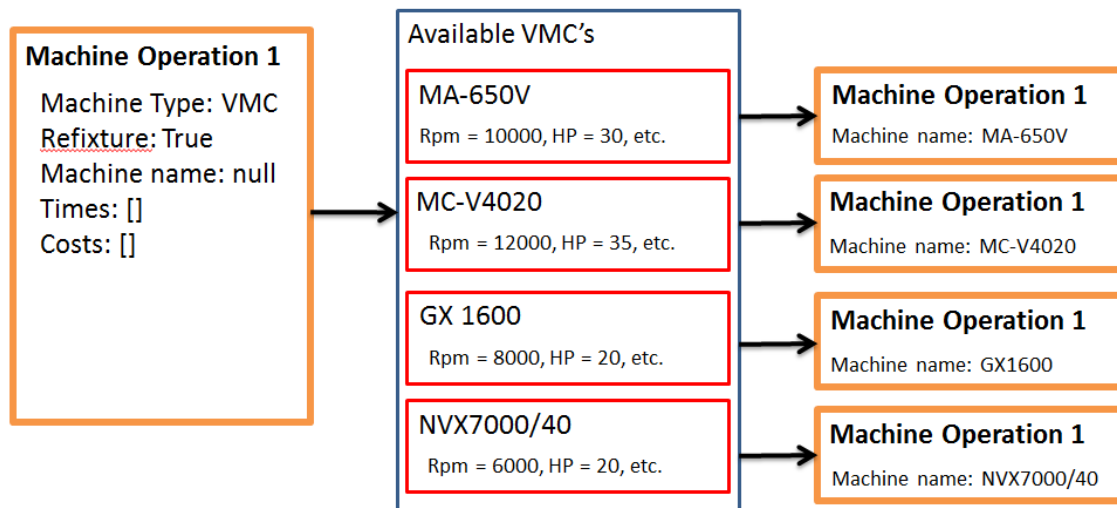


Figure 41: Performing an Operation on all Available Machines

The process described in figure 41 can be carried out for all operations in the machining plan, resulting in a list that describes all options for creating the part, called the machining options lists. The structure of a machining options list can be seen below in figure 42:

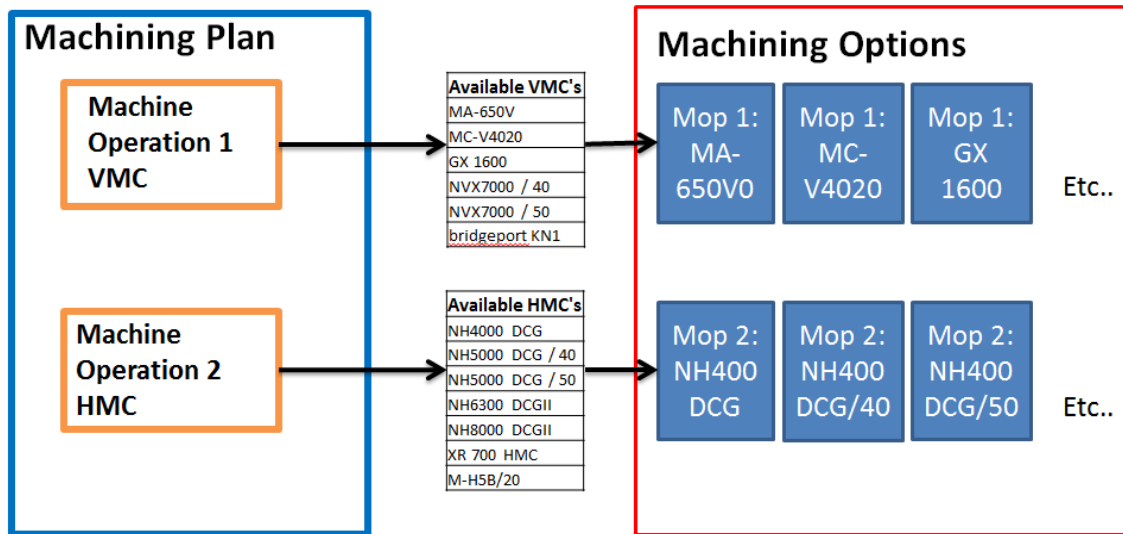


Figure 42: Creation and Structure of a Machining Options List

## Chapter 11: Machining Plan Simplification

The combination of the search that drives AMFA, along with the evaluation of partitions that is able to return a time and cost associated with each operation, is able to return a complete process plan, along with its' associated time and cost. The process plan may or may not be the optimal way to create the part, based on how solutions the search was allowed to create, the type of search used, etc. Whether or not the process plan is optimal, additional processing of the plan is required, to ensure that the feedback to the designer is easily understood and intuitive. The decomposition of parts into partitions is necessary for analysis by AMFA, but it leads to unattractive and complicated machining process plans. However, these process plans can be condensed to give a cleaner and more intuitive feedback to the operator. To understand the need for additional post-processing of process plans, consider the part shown in figure 43:

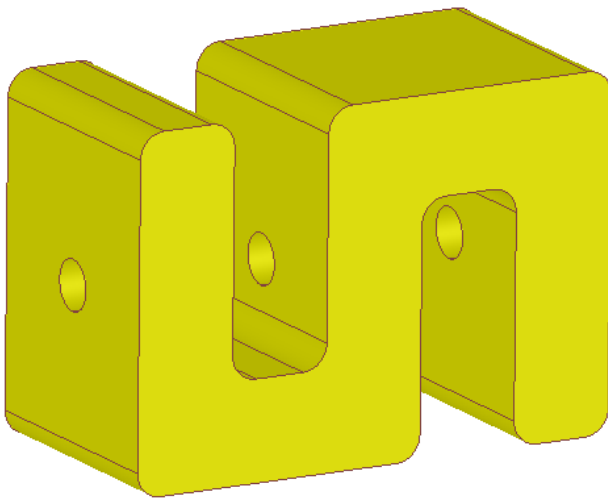


Figure 43: “S” part

This particular part has natural, intuitive separate partitions, as can be seen in the negative shown in figure 44.

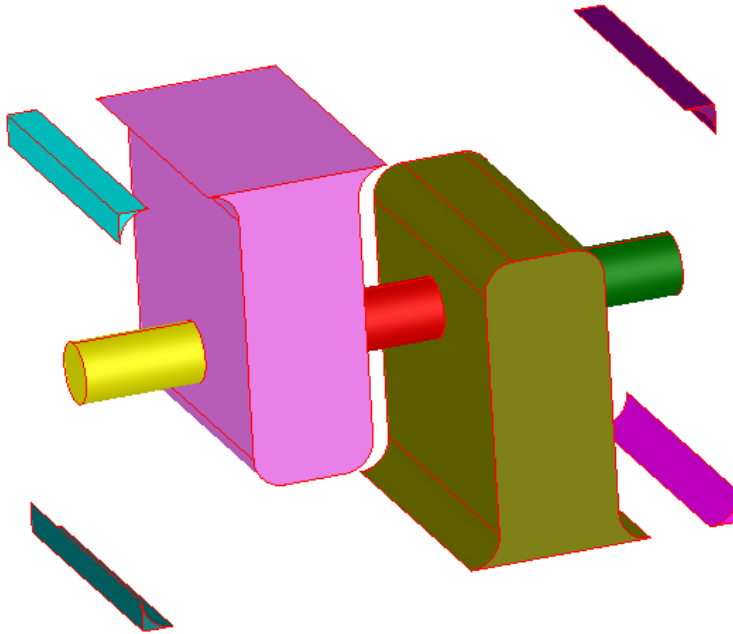


Figure 44: negative of “S”

From figure 44 it can be seen that there are 9 partitions total that make up the negative of this part, therefore there are 9 operations that are created in AMFA to form a complete machining plan. From an intuitive standpoint, however, many of these operations could be combined. Machinists tend to define operations by the number of times they must fixture a part. If more than one “feature” can be machined in that fixture, it is beneficial, as it saves time in the manufacturing of the part. From the human perspective, manufacturing this part would be done in only 3 steps, where each step is a different part fixturing. In each fixture orientation, several features would be removed. The machining plan for this part, produced through human reasoning alone, can be seen below in figure 45.

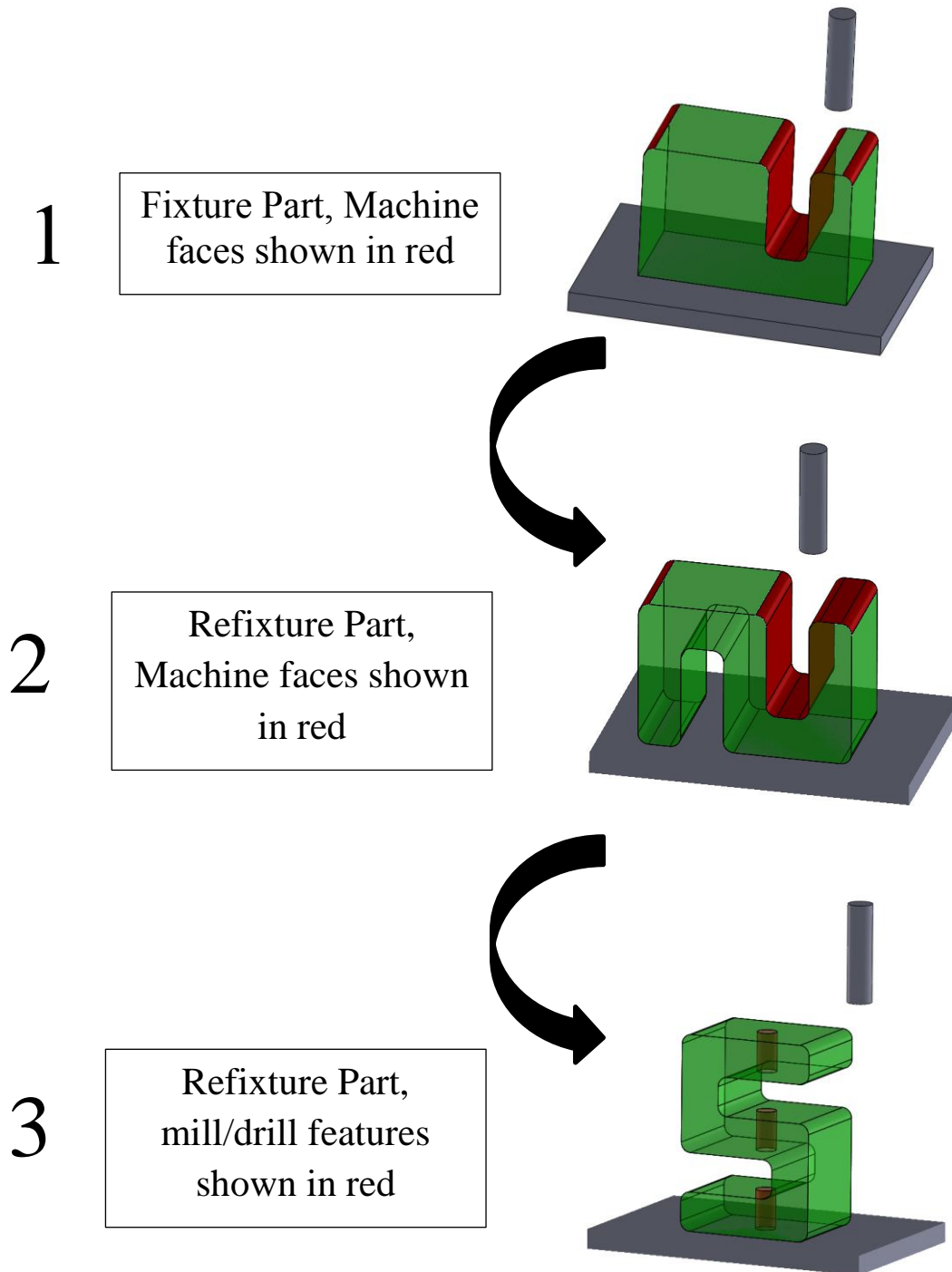


Figure 45: Human generated process plan for "S"

A complete process plan for “S” generated by AMFA would contain not 3 steps, but 9. Not all steps of AMFA will require refixturing. Rather, given a complete search, AMFA too would generate a process plan that would require only 3 total fixtures. This means the reported time and cost of the process plan could be accurate, but the plan would look complicated to the software operator. The process plan generated by AMFA is reported as shown in figure 46 below:

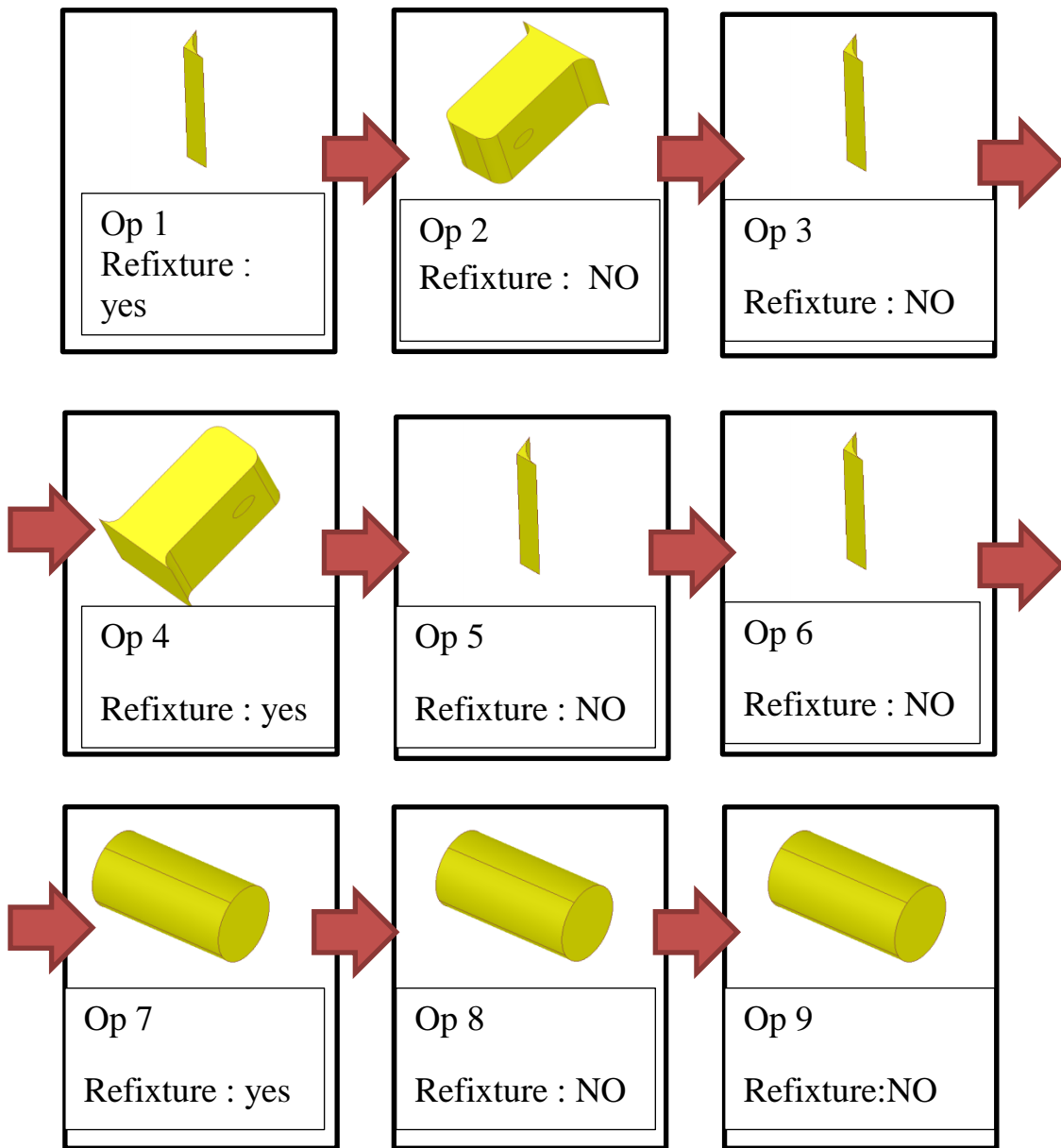


Figure 46: AMFA process plan for “S” part

It is clear from figure 46 and figure 45, that these process plans are equivalent in terms of order of partition removal, but not in fixturing. What is left is to reduce the



number of operations in the AMFA process plan to only those that require refixturing. This would reduce the process plan shown in figure 46 to exactly replicate the process plan generated by the human in figure 45.

In order to collapse the list of machining operations to only the number of unique part setups in the process plan, we make a rule that if an operation has been refixtured, it represents a unique setup, and cannot be combined with the previous operation. This is a simple check to perform, since the machine operations hold a Boolean for refixture, which is set to either true or false. Refixturing is set to true if either the machine has changed or the fixed face has changed, as doing either requires the removal, reorientation / movement of the part, and refixturing. Therefore, refixturing will only ever be false if the operation is in the same machine and the same orientation as the previous operation. If this is the case, the two operations can be combined. An example of two operations that can be combined is shown below in figure 47:

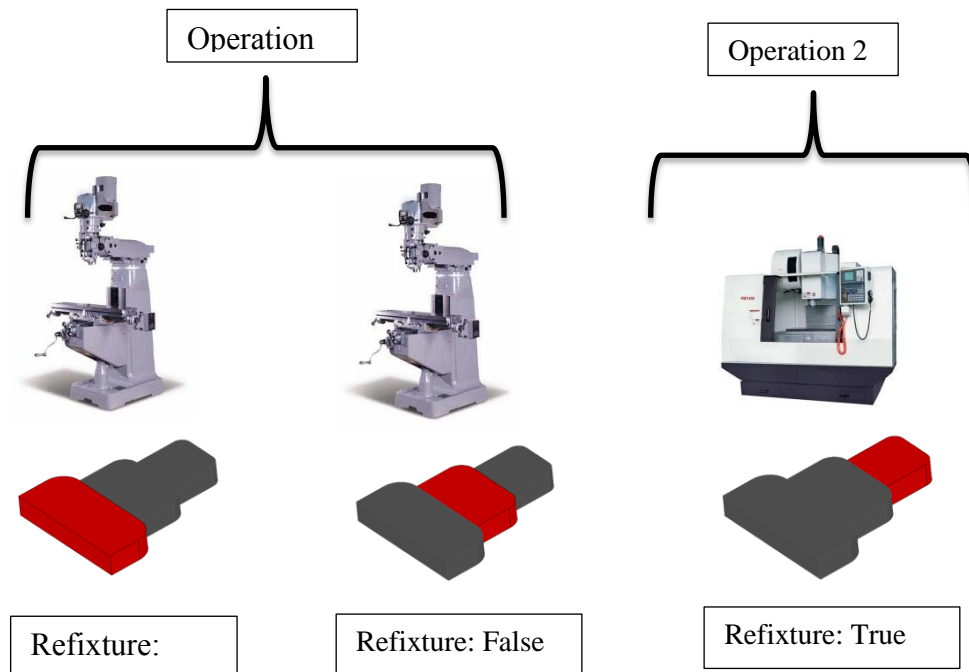


Figure 47: merging two operations

Since this operation now removes two partitions, the time and cost associated with removing each partition needs to be summed to give an accurate account of the requirements of the new combined operation. Since these operations are carried out on the same type of machine, the operations exist on the same set of machines. The operations are combined by merging each machine with the same machine in the other operation, and the values of time and cost are added together element wise, i.e. the 1<sup>st</sup> time value in operation 1 is added to the 1<sup>st</sup> time value in operation 2, etc. Each operation contains 10 points for time and cost, but since the machines are identical, the driving machine parameters at each of these points are identical, so they can be combined.

## **Chapter 12: Candidate Generation**

At this point, the machining options list has been generated, and then reduced to contain only the number of operations that pertain to the number of times the part must be fixtured. To return a complete process plan, however, a selection of a machine for each operation must be performed. In order to return the optimal and most diverse range of solutions to the users, all possible combinations are considered. The generation of candidates begins with the first operation in the machining plan. Each machine entry in the first operation is combined with each entry in the second operation to yield the candidates for performing the first two machining operations. The resulting candidates are then checked for pareto-optimality (described in Chapter 13). The candidates that are determined to be pareto-optimal will then enter another phase of candidate generation with each entry in operation 3. The process repeats until all operations have been analyzed, and each candidate describes a complete process plan. The process of candidate generation can be seen below in figure 48. The format of a generated candidate point can be seen in figure 49.

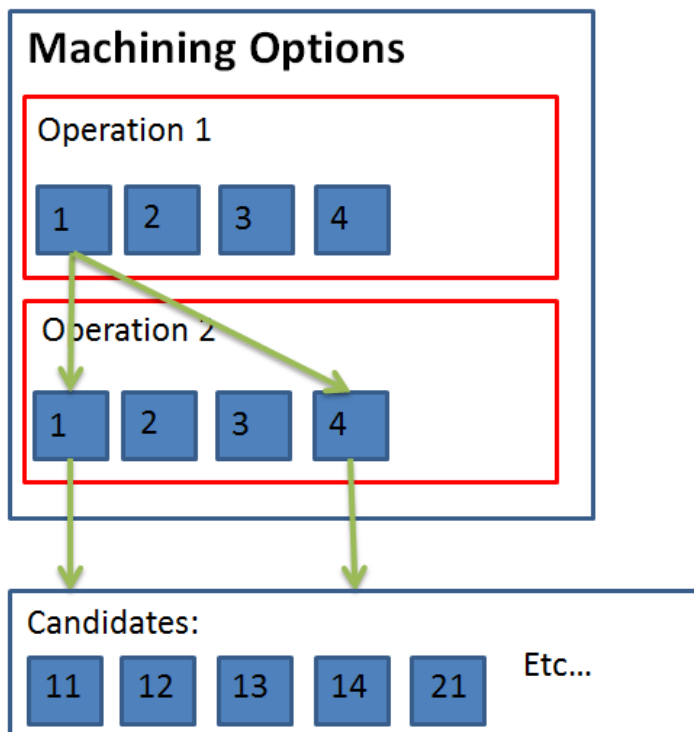


Figure 48: Generation of Candidate Points

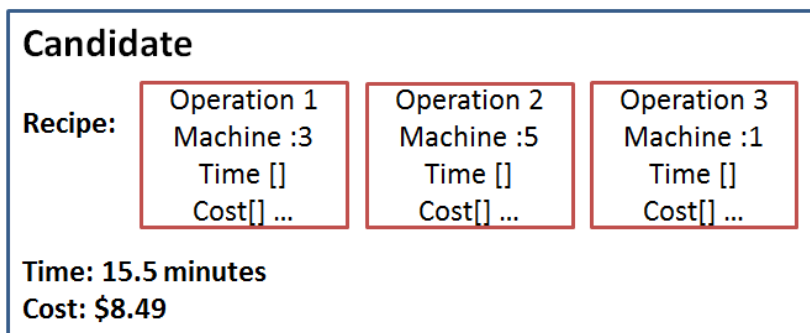


Figure 49: Structure of a Candidate

As can be seen in figure 49, the candidate contains only a single value for time and cost. This is important when the candidate is being checked for optimality against the other candidates. Each operation in the recipe contains a range of times and costs. In order to calculate a single time and cost of each candidate, a single time and cost must be chosen from each operation contained in its recipe. The time and cost chosen from the recipe are picked from the lowest cost machining point. The range of cost values is searched, and the minimum value, along with its corresponding time, are loaded into the candidate as the time and cost of creating this partition. While this gives the user a single value to compare this candidate against others, the full range of values are preserved in the candidate recipe.

## Chapter 13: Candidate Filtering

It is necessary to filter the generated candidates for two reasons. First, we want to eliminate machining plans that are clearly worse than other candidate points, as they are unlikely to be of any interest to the user. Secondly, and possibly more importantly for the software, is it dramatically reduces the number of candidates that must be generated, stored, and presented. For the “S” part, for example, if the process plan cannot be reduced, i.e. the process plan generated by AMFA refixtures the part each time, the reduced process plan will still contain 9 operations. If there are 6 machines available, there will be  $6^9$  candidate points, which is over  $10^7$  candidates. Using candidate filtering, the number of candidate points returned is just over 500.

Candidates are filtered if they are dominated by other candidates. Figure 50 demonstrates a candidate that would be removed due to being dominated by other candidates.

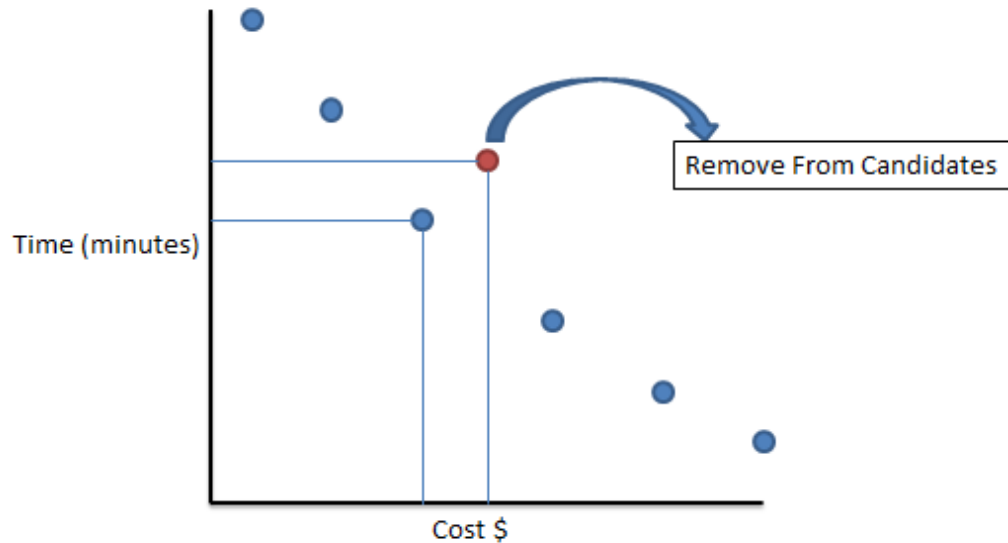


Figure 50: Candidate Filtering by Removing Dominated Candidates

In figure 50, the candidates shown in red has both a higher time and cost than the candidate which can be seen to the lower left. There is no point in keeping a candidate alive that is worse than other possibilities, so this candidate is removed from the list.

## Chapter 14: Process Planning Results

While the development of these methods are relatively recent and still undergoing active development, some preliminary results are available.

### 14.1. STEPPED PART

The stepped feature part shown originally in figure 36, and reproduced in figure 51 below, Is an interesting test case for the software. During processing, the part is split into 3 different partitions, as shown originally in figure 7. The result is that machining plans formed by AMFA will originally contain three operations. It is easy to see that these three partitions can be removed while the part is in a single configuration, and actual tool paths generated for this part were shown in section 51.

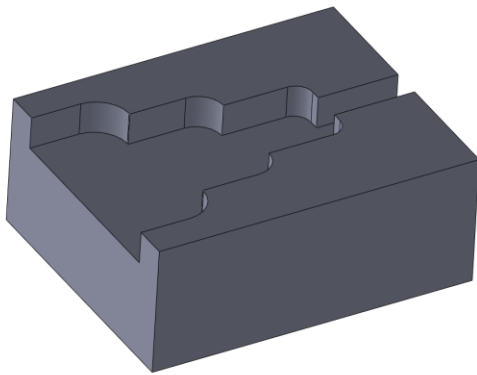


Figure 51: Stepped Feature Part

Since AMFA examines all possibilities for machining a part, a range of machining plan options will be created with respect to fixtures and machine type used. Plans with a different number of fixtures will have greatly different production time and



cost, since fixturing time is a large contribution to the overall time of production, and therefore overall cost as well. Combined with the ability to reduce the number of operations in a process plan as described in Chapter 11, the resulting process plans will contain anywhere from 1-3 operations. An example of each of these cases will be presented briefly, along with a comparison of the general metrics for each plan.

#### **14.1.1 3 Operations, 3 Fixtures**

If we were unable to combine any of the operations in the machining plan into a single operation, we are left with 3 operations in the process plan. For each of these operations we consider the use of all available machines. The resulting pareto-optimal points returned as machining options can be seen in figure 52. The details of each of these candidates can be seen in table 5.

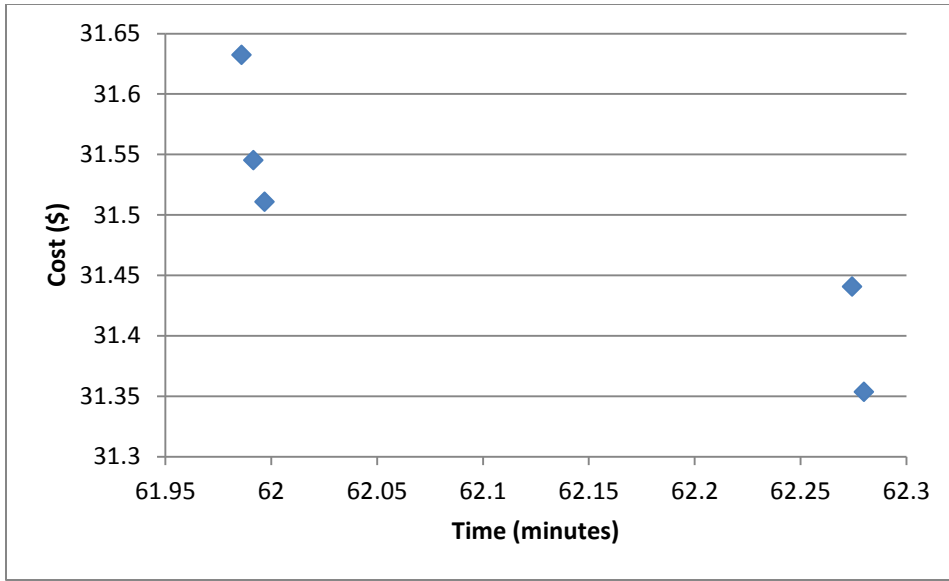


Figure 52: 3 Operations for Stepped Feature

Cost	Time (mins)	Machine 1	Machine 2	Machine 3
61.99683	31.51063963	MA-650V	XR 700 HMC	bridgeport KN1
61.98605	31.63226586	MC-V4020	XR 700 HMC	MA-650V
61.99164	31.54515846	MC-V4020	XR 700 HMC	bridgeport KN1
62.27443	31.44055201	bridgeport KN1	XR 700 HMC	MA-650V
62.28002	31.3534446	bridgeport KN1	XR 700 HMC	bridgeport KN1

Table 5: Details for Candidate Points for 3 Operations for Stepped Feature

This particular plan involved using a VMC for machining the first operation, an HMC for the second operation, and a VMC again for the final operation. Although there are 7 HMC's available, only 1 was selected in this list of pareto-optimal points. From this we can conclude that this machine had a greater performance at a lower cost than all

of the other HMC's available. At this point it is useful to present all of the machine data used in these simulations. This data is shown in table 6.

model Id (String)	work table Traverse Rate (mm/min)	spindle Power Rating (hp)	spindle Orientation (String)	billing Rate (\$/hr)	rpm	tct
MA-650V	30000	11	vertical	25	6000	2
XR 700 HMC	48000	25	horizontal	27	12000	2.2
M-H5B/20	40000	20	horizontal	28	12000	3
MC-V4020	35991.8	15	vertical	31	15000	2.5
GX 1600	36000	35	vertical	35	10000	2.3
NVX7000 / 40	20000	11	vertical	33	20000	2
NVX7000 / 50	30000	15	vertical	40	8000	2
NH4000 DCG	25000	24.7	horizontal	36	14000	2.8
NH5000 DCG / 40	30000	25	horizontal	45	14000	3
NH5000 DCG / 50	30000	25	horizontal	40	8000	2.5
NH6300 DCGII	30000	25	horizontal	43	10000	3
NH8000 DCGII	60000	30	horizontal	100	10000	1.9
bridgeport KN1	5000	3	vertical	10	3000	30

Table 6: Machine Data Available to AMFA

Examining the machine data shown in table 6 makes the resulting candidate points shown in figure 52 easier to understand. The “XR700 HMC” dominated all other HMC's because of its relatively high motor power, high spindle speed, and low billing rate. This gives the machine the ability to machine quickly and at a lower cost. Since the VMC machines are more diverse, several VMC candidates are maintained in the pareto-optimal set. The first three candidate plans, in the left group of points in figure 52, are created using more expensive machines for the first (largest) partition. This leads to the

candidates having an overall lower time and higher cost than the group on the right, which use a simple, but affordable mill to remove the large partition. This simple machine, due to its exceptionally low billing rate, is used in many of the pareto-optimal points. The billing rate on this machine was created to represent the “collecting dust in the corner” machine, where using that machine may translate to the lowest cost manufacturing, even though the machinist typically would not consider it.

#### 14.1.2. 2 Operations, 2 Fixtures

If the machining plan generated by AMFA contains two adjacent operations that can be merged, the resulting process plans will contain only two operations. The resulting pareto-optimal candidates can be seen below in figure 53, and are presented with their associated machine selections in table 7.

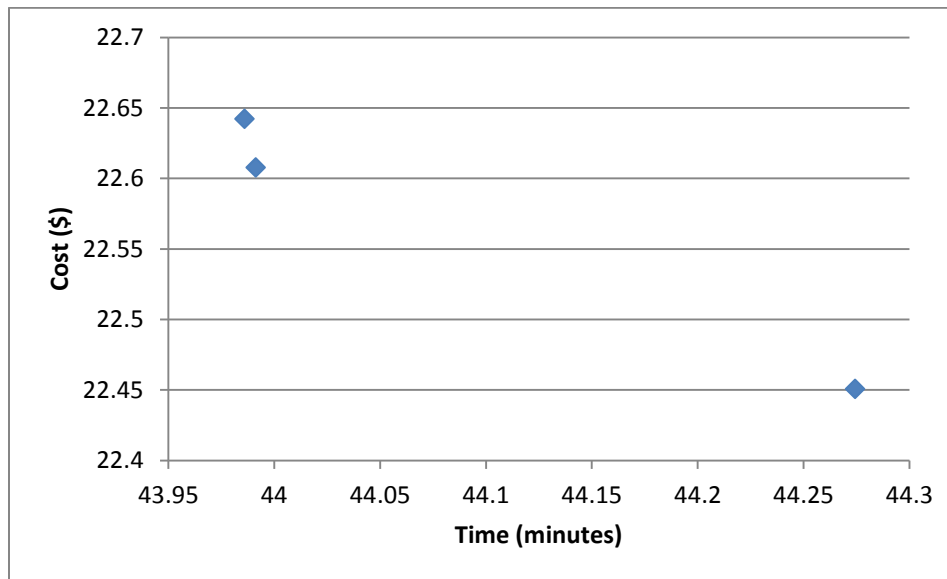


Figure 53: 2 Operations for Stepped Feature

Cost	Time (mins)	Machine 1	Machine 2
43.99124	22.60773266	MA-650V	XR 700 HMC
43.98605	22.64225148	MC-V4020	XR 700 HMC
44.27443	22.45053763	bridgeport KN1	XR 700 HMC

Table 7: Details for Candidate Points for 2 Operations on Stepped Feature Part

The existence of only 3 candidate points for this machining plan is encouraging, as it has reduced the 42 possibilities to a number of candidates that is reasonable to expect the operator of the software to easily interpret and understand. Once again, we can see that a small, cheap machine, and two faster, more expensive machines have been presented as options. Like the 3 operation case, the second operation required an HMC, and the XR700 beat out all other machines for dominance. What this means is that the second operation had no effect on which candidates were pareto-optimal, meaning only 3 candidates survived after operation 1.

#### 14.1.3. 1 Operation, 1 Fixture

If we examine a machining plan where all operations can be merged into 1 operation, we can confirm that only 3 candidates survive. The candidates are shown in figure 54, and presented in table 8.

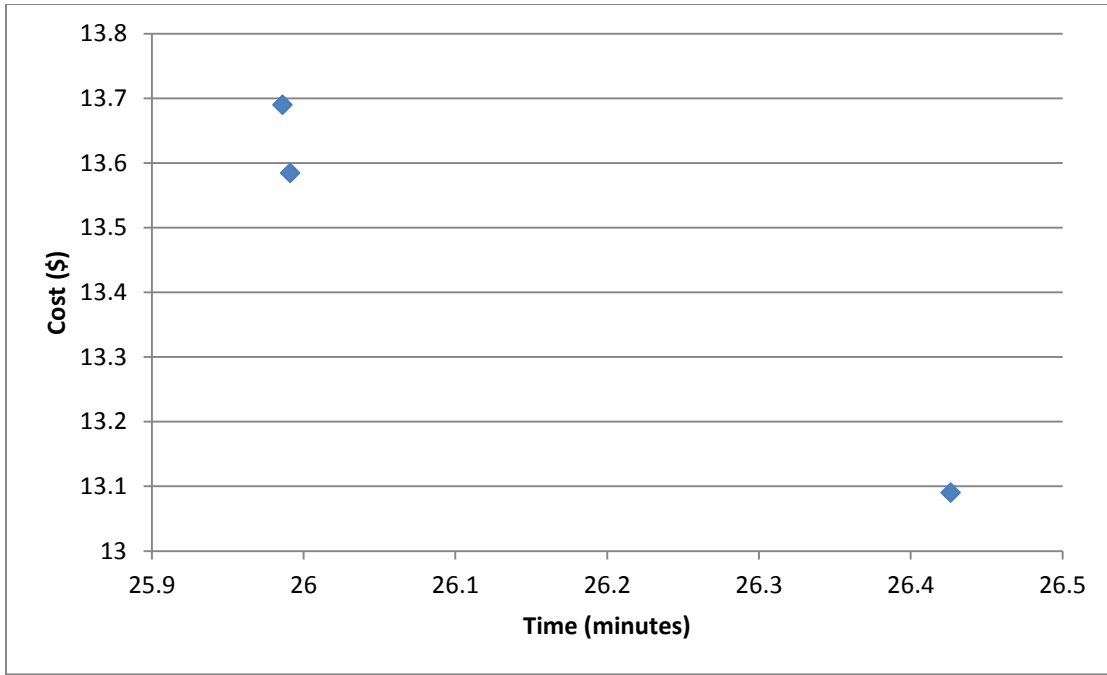


Figure 54: 1 Operation for Stepped Feature

Cost	Time (mins)	Machine 1
25.99124	13.58413084	MA-650V
25.98605	13.68945512	MC-V4020
26.42635	13.0898207	bridgeport KN1

Table 8: Details for Candidate Points for 1 Operation on Stepped Feature Part

Examining figures 53 and 54, we can see that the layout of the candidates is exactly the same, just time shifted, as there is only one fixture time for the single operation case, rather than 2.

If these three machining plans are plotted together, it is easy to recognize the benefit of minimizing the number of operations required to make a part. The plot showing all three previously discussed plans is shown in figure 55:

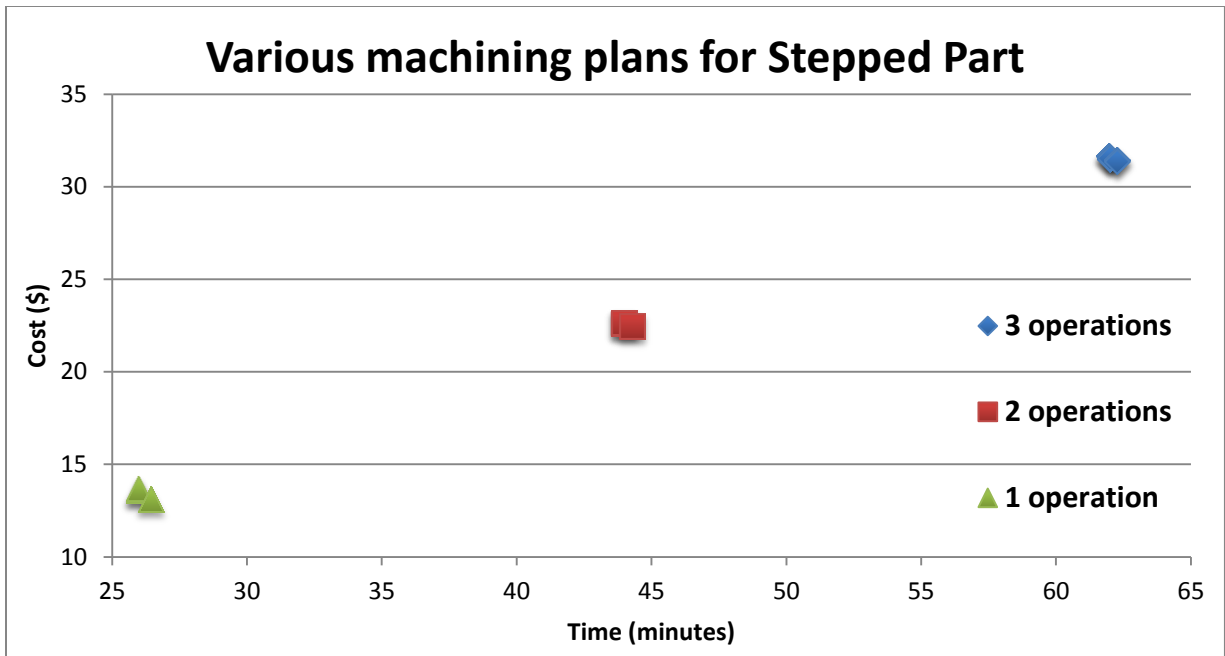


Figure 55: Variety of Machining Plans for Stepped Feature Part

Each cluster shown in figure 55 is the set of pareto-optimal process plans based on one machining plan, where a different number of total operations were found for each machining plan. The “1 operation” points are shown in figure 54, the “2 operation” points are shown in figure 53, and the “3 operation” points are shown in figure 52.

#### 14.2. “S” PART

The generation of pareto-optimal process plans becomes more interesting when a higher number of operations and fixtures are required. Consider again the “S” shaped part shown in figure 43, with the negative shown in figure 44. As previously discussed, it is possible to create this part with only 3 fixtures, and 3 operations in the reduced process plan. Unfortunately, at this point AMFA is using a blind search to generate machining plans, so while it find options, it rarely produces the minimum number of fixtures

required to create the part. Although the machining plans evaluated here are not the optimal way to make the “S” part (the resulting time will be higher than it could be with a better machining plan), it will still allow for interesting analysis of the machine selection impact and the pareto-optimal options for machining a several-step part.

Each of the plans presented here contains 8 operations in the reduced process plan. The difference in these plans comes from the order of operations, and the number of each type of machine used. VMC’s and HMC’s use a slightly different fixture time, leading to small differences in the manufacturing time depending on which type of machine is used.

#### **14.2.1 Fillets First**

The process plan presented here had the removal of the 4 small corner fillets first. These parts are very small, and thus require a high rpm on the machine to cut well. The candidates that are presented for machining the “S” part for this plan are shown in figure 56, and details for a select few of these points are selected in table 9:



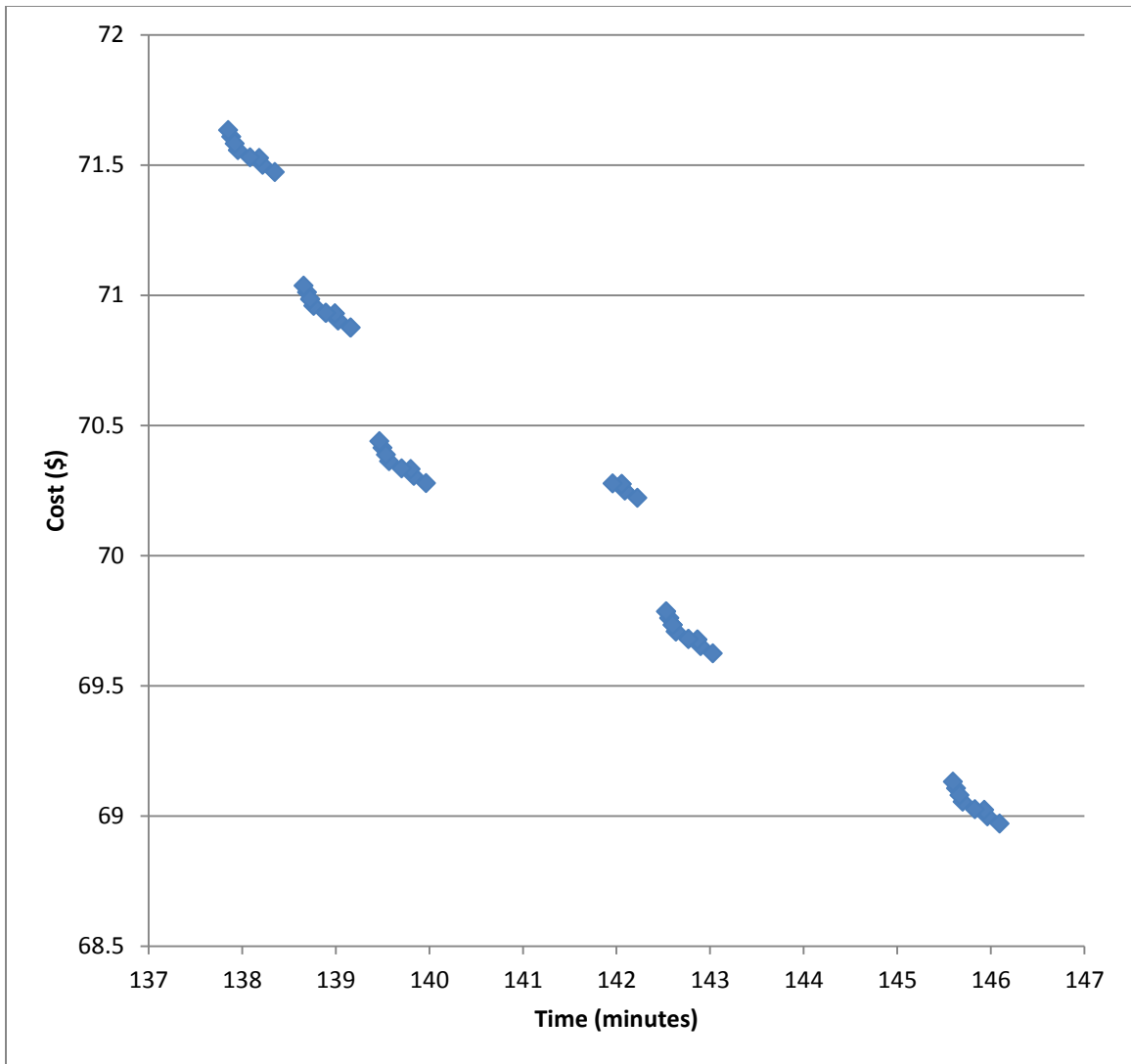


Figure 56: “S” part, 8 Operations

Cost	Time (mins)	Machine 1-4	Machine 5	Machine 6	Machine 7	Machine 8
139.5715518	70.36197006	NVX7000 / 40	MA-650V	MA-650V	MA-650V	MA-650V
138.7633021	70.95927243	NVX7000 / 40	MA-650V	MA-650V	MA-650V	MC-V4020
142.636082	69.70814101	NVX7000 / 40	MA-650V	MA-650V	MA-650V	bridgeport KN1
139.5019951	70.41337301	NVX7000 / 40	MA-650V	MA-650V	MC-V4020	MA-650V
142.5665252	69.75954395	NVX7000 / 40	MA-650V	MA-650V	MC-V4020	bridgeport KN1
139.8352807	70.30570251	NVX7000 / 40	MA-650V	MA-650V	bridgeport KN1	MA-650V
139.0270309	70.90300488	NVX7000 / 40	MA-650V	MA-650V	bridgeport KN1	MC-V4020
142.8998109	69.65187346	NVX7000 / 40	MA-650V	MA-650V	bridgeport KN1	bridgeport KN1
138.7633021	70.95927243	NVX7000 / 40	MA-650V	MC-V4020	MA-650V	MA-650V
137.9550523	71.5565748	NVX7000 / 40	MA-650V	MC-V4020	MA-650V	MC-V4020
138.6937453	71.01067538	NVX7000 / 40	MA-650V	MC-V4020	MC-V4020	MA-650V
137.8854955	71.60797775	NVX7000 / 40	MA-650V	MC-V4020	MC-V4020	MC-V4020
139.0270309	70.90300488	NVX7000 / 40	MA-650V	MC-V4020	bridgeport KN1	MA-650V
138.2187811	71.50030725	NVX7000 / 40	MA-650V	MC-V4020	bridgeport KN1	MC-V4020
142.0915611	70.24917583	NVX7000 / 40	MA-650V	MC-V4020	bridgeport KN1	bridgeport KN1
142.636082	69.70814101	NVX7000 / 40	MA-650V	bridgeport KN1	MA-650V	MA-650V
145.7006122	69.05431195	NVX7000 / 40	MA-650V	bridgeport KN1	MA-650V	bridgeport KN1
142.5665252	69.75954395	NVX7000 / 40	MA-650V	bridgeport KN1	MC-V4020	MA-650V

Table 9: “S” part, 8 Operations

In table 9, one column is labeled Machine 1-4, and is filled entirely with the NVX7000/40. As was mentioned previously, the first 4 operations in this machining plan are removing the corner fillets, which require a very high spindle speed. This machine

has the highest spindle speed in the database, along with a reasonable billing rate, meaning that it dominates all other candidates for the machining of these 4 partitions. The remaining partition are larger volumes, where other machines become pareto-optimal as well.

The overall shape of the pareto-optimal solution space shown in figure 56 is intuitive, but the existence of point clusters is harder to understand. If we make enlargements of two of these clusters, the fastest (upper left) and slowest (lower right) clusters, we can learn more about why the process plan options become grouped in this way.

Machine 5	Machine 6	Machine 7	Machine 8
MA-650V	MA-650V	MA-650V	MC-V4020
MA-650V	MC-V4020	MA-650V	MA-650V
MA-650V	MC-V4020	MA-650V	MC-V4020
MA-650V	MC-V4020	MC-V4020	MA-650V
MA-650V	MC-V4020	MC-V4020	MC-V4020
MA-650V	MC-V4020	bridgeport KN1	MC-V4020
MC-V4020	MA-650V	MA-650V	MC-V4020
MC-V4020	MC-V4020	MA-650V	MC-V4020
MC-V4020	MC-V4020	MC-V4020	MC-V4020
bridgeport KN1	MC-V4020	bridgeport KN1	MC-V4020

Table 10 : Fast and Costly Cluster

Machine 5	Machine 6	Machine 7	Machine 8
MA-650V	bridgeport KN1	MA-650V	bridgeport KN1
MA-650V	bridgeport KN1	bridgeport KN1	bridgeport KN1
MC-V4020	bridgeport KN1	MA-650V	bridgeport KN1
bridgeport KN1	bridgeport KN1	MA-650V	bridgeport KN1
bridgeport KN1	bridgeport KN1	bridgeport KN1	bridgeport KN1

Table 11: Slow and Cheap Cluster

Examination of the data in table 10 and 11 shows that each cluster is merely a single machine being dominant. In this case, there are 6 machines under consideration, as all partitions have been removed by VMC's. There are 6 clusters shown in figure 56. Each cluster is dominated by one of the 6 machines. It is logical that if a single machine was used to machine all partitions, the 6 resulting candidates would create the same shape as the 6 clusters in figure 56. The fast and costly cluster contains this plan, where MC-V4020 machine has been used for all operations (besides the first 4 fillets). Likewise, the slow and cheap cluster contains a candidate where the Bridgeport KN1, our least capable but most affordable machine, is used for all operations.

Varying the machining plan still produces different overall candidates and values. Three such machining plans have been generated and the pareto-optimal candidates for each plan are shown in figure 57:

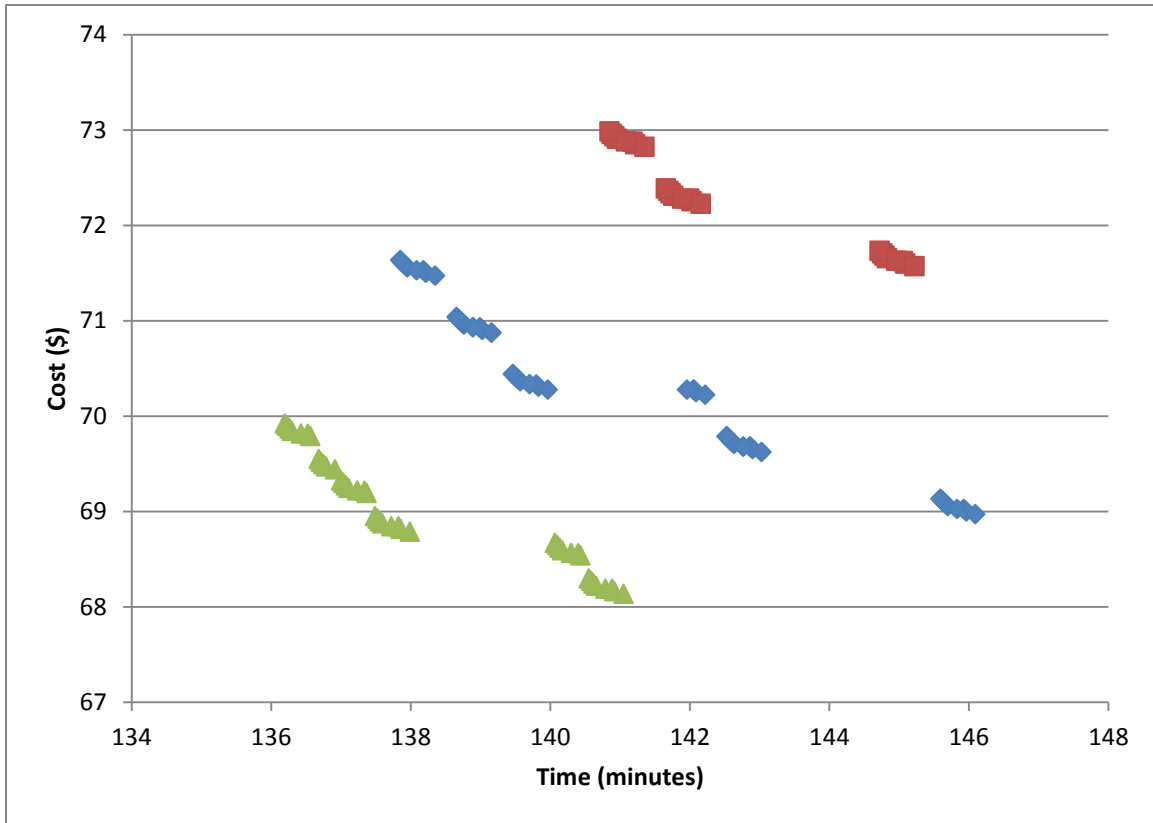


Figure 57: Machine-Specific Candidates for 3 Different Machining Plans

Note that for two of the machining plans shown in figure 57, there are 6 clusters of points, as discussed previously. It appears for the third case that there are only 3 clusters. In this machining plan, an HMC was used heavily, for which only one machine was ever selected as optimal. The remaining operations performed on a VMC contributed less to the overall clustering, but if these points are examined closely, 6 clusters can still be made out and clearly defined.

## **Chapter 15: Discussion and Future Work**

As many of the techniques presented here are unique and undergoing current development, many of them still face challenges and assumptions. The tool selection technique operates with tool diameter only, and is currently being expanded to include considerations of tool length, number of flutes, and tool material. The machining model currently is not able to reason about part quality or tolerance, but it is currently being researched. The tool wear model is still in early stages, and will require testing to gather tooling constants and further model manipulation to instill confidence in the results.

The ability to consider all machines available to the user to create a part is a powerful and useful tool, which could be enhanced if the ability to track the “up-time” of the machine was recorded. This would allow process plans which call for the use of machines which are already in high demand to be lower, and machines that are used less to rated higher. While this is not a difficult model to implement in the software, gathering metrics for tracking the machine use and validating the model would be a much more difficult and long process.

Future development of AMFA will see the addition of several machining considerations and parameters that will greatly increase the applicability of this software for industrial manufacturing. Currently the addition of surface finish and tolerance considerations in the machining model is being researched. The ability to reason about these parameters would allow the user to choose plans based on a third parameter, and would allow users to quickly and easily evaluate the increased cost of machining for tighter tolerances or better surface finish.

The types of machines considered in the software are being expanded, and will eventually include all machines found in a typical manufacturing foundry. Once AMFA

is able to consider all machines, any solid model could be processed, and it is likely that the software could suggest ways to create the part that would not otherwise be considered (i.e. sheet cut 2 pieces and weld rather than machine), and show the user, in terms of time, cost, and part quality, why a certain way of making a part is superior.

Industry professionals who have been shown the work presented here are impressed and excited by the capability of our software. While what AMFA is able to do is already cutting-edge, it is the possibility for the future development of AMFA that gathers the most interest. If the search process in AMFA is able to always find an optimal machining plan, and we can execute that machining plan on all machines available, and return to the user a set of options for creating the part, we will have achieved the goal of recreating the “mind of the machinist”, except the process has been completed in only minutes, instead of hours or days. In addition, the operator is able to choose plans based on machine preference, time constraints, etc., all parameters that until now were not easily defined and evaluated in a manufacturing setting.

It is clear that AMFA has the potential to not only replace hours of human labor, but grant a manufacturing facility an unprecedented understanding of the cost of manufacturing a part, the implications of varying process plans or machining parameters, and how to optimize their facility for the lowest cost, the highest throughput, or the highest quality parts.

## Bibliography

- [1] "STEP Application Protocols." [Online]. Available: [http://www.steptools.com/library/standard/step\\_2.html](http://www.steptools.com/library/standard/step_2.html). [Accessed: 2012].
- [2] A. A. Eftekharian; M. I. Campbell, "Convex Decomposition of 3D Solid Models for Automated Manufacturing Process Planning Applications," in *ASME International Design Engineering Technical Conferences IDETC2, Chicago, Illinois, USA, 2012.*, 2012.
- [3] M. I. C. and C. F. W. Fu, P. Radhakrishnan, A. A. Eftekharian, "A GRAPH GRAMMAR BASED APPROACH TO AUTOMATED MANUFACTURING PLANNING," in *ASME International Design Engineering Technical Conferences IDETC, Chicago, IL, USA, 2012.*, 2012.
- [4] R. Enparantza, O. Revilla, and A. Azkarate, "A life cycle cost calculation and management system for machine tools," *13th CIRP International Conference on Life Cycle Engineering*, pp. 717-722, 2006.
- [5] M. BALA and T.-C. CHANG, "Automatic cutter selection and optimal cutter path generation for prismatic parts," *International Journal of Production Research*, vol. 29, no. 11, pp. 2163-2176, Nov. 1991.
- [6] D. VEERAMANI and Y.-S. GAU, "Cutter-path generation using multiple cutting-tool sizes for 2-1/2D pocket machining," *IIE Transactions*, vol. 32, no. 7, pp. 661-675, Jul. 2000.
- [7] D. E. R. C. T. Lim, J. Corney, "Exact Tool Sizing for Feature Accessibility," *International Journal of Advanced Manufacturing Technology*, vol. 16, no. 11, pp. 791 - 802, 2000.
- [8] J. M. R. T. Lim, "Optimising Automatic Tool Selection for 2 1/2D Components," in *ASME 2000 Design Engineering Technical Conferences and Computers and Information in Engineering Conference Baltimore, Maryland, September 10-13, 2000*, 2000.
- [9] I. D Carpenter and P. G Maropoulos, "A flexible tool selection decision support system for milling operations," *Journal of Materials Processing Technology*, vol. 107, no. 1-3, 2000.
- [10] "Apriori Product Cost Management." [Online]. Available: [www.apriori.com](http://www.apriori.com). [Accessed: 2012].



- [11] "COSTIMATOR" [www.costimator.com](http://www.costimator.com)
- [12] J. Hoschek, "Offset curves in the plane," *Computer-Aided Design*, vol. 17, no. 2, pp. 77-82, Mar. 1985.
- [13] "Open CASCADE." p. 2001. < need url >
- [14] J. R. Walker, *Machining Fundamentals*. Goodheart-Willcox, 2004, p. 640.
- [15] "FeatureCAM" [www.featurecam.com](http://www.featurecam.com)
- [16] I. Zeid, *CAD/Cam Theory and Practice*. 1991.
- [17] R. R. Martin and P. C. Stephenson, "Sweeping of three-dimensional objects," *Computer-Aided Design*, vol. 22, no. 4, pp. 223-234, May 1990.
- [18] H. El-Hofy, *Fundamentals of machining processes: conventional and nonconventional processes, Volume 978, Issues 0-7289*. CRC/Taylor & Francis, 2006, p. 452.
- [19] S. Kalpakjian and S. R. Schmid, *Manufacturing processes for engineering materials*. Pearson Education, 2008, p. 1018.
- [20] J. Gu, G. Barber, S. Tung, and R.-J. Gu, "Tool life and wear mechanism of uncoated and coated milling inserts," *Wear*, vol. 225-229, pp. 273-284, Apr. 1999.
- [21] J. Z. Zhang, J. C. Chen, and E. D. Kirby, "Surface roughness optimization in an end-milling operation using the Taguchi design method," *Journal of Materials Processing Technology*, vol. 184, no. 1-3, pp. 233-239, Apr. 2007.
- [22] J. M. Vieira, A. R. Machado, and E. O. Ezugwu, "Performance of cutting fluids during face milling of steels," *Journal of Materials Processing Technology*, vol. 116, no. 2-3, pp. 244-251, Oct. 2001.
- [23] R. R. Kibbe, R. O. Meyer, J. E. Neely, and W. T. White, *Machine Tool Practices*. Prentice Hall, 2009, p. 800.